UNIVERSITATEA *SPIRU HARET*

CORNEL POPA

LOGICĂ ȘI METALOGICĂ

I

EDITURA FUNDAȚIEI *ROMÂNIA DE MÂINE*

© Editura Fundaţiei „România de Mâine", 2000
ISBN 973-582-255-5

Redactor: OCTAVIAN CHEŢAN
Tehnoredactor: BRÎNDUŞA DINESCU
Coperta: RĂZVAN-MIHAI SANDER

Bun de tipar: 12-06-2000; Coli de tipar: 24,5
Format: 16/61x86

Editura şi Tipografia Fundaţiei „România de Mâine"
Splaiul Independenţei, nr. 313, Bucureşti, sector 6, O.P. 78
Tel.: 410.43.80, Fax: 411.33.84; www.SpiruHaret.ro

UNIVERSITATEA „SPIRU HARET"

FACULTATEA DE FILOSOFIE ŞI JURNALISTICĂ

Prof. univ. **CORNEL POPA**

LOGICĂ ŞI METALOGICĂ

I

EDITURA FUNDAŢIEI „ROMÂNIA DE MÂINE"
Bucureşti, 2000

CUPRINS

Cuprins

INTRODUCERE. OBIECTUL LOGICII 5

1. Teoria logică . Limbajul unei teorii logice 6
2. Principiile logice şi tipologia sistemelor logice 9
   2.1. Valenţa sistemelor logice 10
   2.2 Principiul consistenţei sau noncontradicţiei 12
   2.3. Principiul excluderii celei de a n+1 valori 13
   2.4 Principiul constanţei în valorizarea unei propoziţii 15
   2.5. Principiul verifuncţionalităţii 16
3. Consecinţe logice 17
4. Logica şi legile gândirii 20
5. Logica şi legile existenţei 23
6. Adevărul, logica şi teoria cunoaşterii 25
7. Logica şi teoria limbajelor 27
8. Gândirea logică, timpul şi activităţile umane 29
9. Descriptiv şi normativ în logică 31
10. Niveluri ale demersului logic şi accepţii ale termenului logică 33
11. „Harta logicii" şi direcţii ale cercetării logice contemporane 37
12. Concluzii 39
13. Teme şi probleme de dezbatere.Posibile subiecte de examen 41
14. Bibliografie 41

Secţiunea I. LOGICA PROPOZIŢIILOR 43

Cap. 1. SINTAXA ŞI SEMANTICA LOGICII PROPOZIŢIILOR 45

1. Propoziţie şi semn propoziţional 45
2. Limbajul logicii propoziţiilor 46
3. Operaţii şi funcţii logice. Evaluări semantice 49
4. Operaţiile logice şi limba naturală 52

Cap 2. TEORIA FUNCŢIILOR DE ADEVĂR 57

1. Formule şi funcţii logice. Tabloul funcţiilor logice bivalente 57
2. Tabloul funcţiilor logice bivalente şi teoria mulţimilor 59
3. Principiul dualităţii 63
4. Algebra funcţiilor de adevăr 65
5. Interdefinibilitatea şi reducerea operaţiilor logice 66
6. Implicaţii şi consecinţe 68
7. Pătratul logic şi tabloul funcţiilor logice bivalente 70

Cap. 3. METODE DE DECIZIE ÎN LOGICA PROPOZIŢIILOR 74

1. Realizabilitate şi validitate 74
2. Metode semantice 75
   2.1. Tipologia metodelor de decizie 75
   2.2. Metoda matriceală 76
   2.3. Metoda lui W. O. Quine 78
   2.4. Metoda tablourilor semantice 79
   2.5. Metoda arborilor de decizie 83

Cap. 4. CALCULUL SECVENŢIAL 105

1. Consecinţa logică şi calculul secvenţial 105
2. Regulile calculului secvenţial 107

Cap. 5. FORME NORMALE. DECIZIA PRIN FORME NORMALE 115

1. Echivalenţele logic-propoziţionale şi normalizarea 116
2. Formele normale conjunctive 118
3. Formele normale disjunctive 120
4. Formele normale disjunctive perfecte 122
5. Formele normale conjunctive perfecte 125
6. Formele normale, teoria modelelor şi perspectiva sistemică 127

Cap. 6. METODA REZOLUŢIEI ÎN LOGICA PROPOZIŢIILOR 131

1. Schemele de inferenţă şi principiul rezoluţiei 131
2. Teorema rezolventului 135
3. Derivare rezolutivă şi respingere rezolutivă 139
4. Teorema derivării rezolutive 143
5. Principiul rezoluţiei şi metoda rezoluţiei 145
6. Strategii rezolutive 147
7. Rezoluţia teleologică 157
8. Arborii închişi şi respingerile rezolutive 163

9. Rezoluţia duală, diviziunea şi minimizarea 166
9.1. Rezolvent dual şi rezoluţie duală. Specii de rezoluţii duale 166
9.2. Principiul rezoluţiei duale 169
9.3. Derivare rezolutivă duală 171
9.4. Teorema derivării rezolutive duale 173
9.5. O nouă interpretare a principiului rezoluţiei duale 174
9.6. Formule reduse clauze factor şi minimizări 179
9.7. Rezolvenţi duali simetrici 185
9.8. Rezoluţia duală simetrică şi minimizarea 187
9.9. Dualitatea rezoluţiilor Robinson şi Maslov 189

Cap. 7. METODA EXTINDERII CONSERVATIVE A INTERPRETĂRILOR PARŢIALE 199

Cap. 8. METODA DAVIS–PUTNAM 211

1. Regulile metodei Davis–Putnam 211
2. Teorema scindării (Splitting Rule) 211
3. Teorema scindării şi principiul rezoluţiei 221
4. Formele normale disjunctive perfecte şi metoda Davis–Putnam duală 227
5. Teorema Davis–Putnam duală 230
6. Metoda Davis–Putnam duală 231

Cap. 9. FORMELE NORMALE, METODA REZOLUŢIEI ŞI ECUAŢIILE LOGICE 237

1. Formele normale şi rezolvarea de probleme 237
2. Clauzele disjunctive, arborii şi-sau şi instrucţiunile Prolog 243
3. Logica propoziţiilor şi programarea logică 248

Secţiunea a II-a. LOGICA PREDICATELOR ŞI BAZELE DE CUNOŞTINŢE 253

Cap.1. SINTAXA ŞI SEMANTICA LOGICII PREDICATELOR 255

1. Întrebări şi probleme 255
2. Limbajul logicii predicatelor 258
3. Semantica logicii predicatelor 264
4. Predicate şi propoziţii. Întrebări şi răspunsuri 267

Cap. 2. METODE DE DECIZIE ÎN LOGICA PREDICATELOR 272

1. Metoda tablourilor semantice 272

2. Metoda arborilor de decizie 276

3. Calculul secvenţial în logica predicatelor 288
4. Metoda rezoluţiei 297
5. Strategii rezolutive în logica predicatelor 342
5.1. Rezoluţia liniară 344
5.2. Strategia rezoluţiei semantice 350
Concluzii 355
Bibliografie selectivă 357

ANEXE – SUBIECTE DE EXAMEN 359

Anexa A. Subiecte teoretice de logica propoziţiilor 359
Anexa B. Exerciţii de logica propoziţiilor 361
Anexa C. Exerciţii de formalizare. Definiţii de predicate şi propoziţii din predicate primare date 363
Anexa D. Exerciţii de decodificare 368
Anexa E. Subiecte de silogistică rezolvate prin logica predicatelor 376

CONTENT

INTRODUCTION. THE OBJECT OF LOGIC 5

1. Logical Theory. The Language of a Logical Theory 6
2. Logic Principle and the Typology of the Logical Systems 9
   2.1. Valences and Logical Systems 10
   2.2 Consistency and Noncontradiction Principle 12
   2.3. The Exclusion of the n+1 Value Principle 13
   2.4 Constancy in Value Assignement Principle 15
   2.5. Truthfunctionality Principle 16
3. Logical Consequence 17
4. Logic and the Laws of Thought 20
5. Logic, Ontology and Natural Laws 23
6. Truth, Logic and Theory of Knowledge 25
7. Logic, Natural and Formal Languages 27
8. Logical Thinking, Time an Human Activities 29
9. Descriptive and Normative Dimensions of Logic 31
10. Levels of the Logical Approaches and Definition of Logic 33
11. „The Chart of Logic" and the Directions of the Contemporary Logical Inquiry 37
12. Conclusions 39
13. Subject proposals for essays and debates. Subjects for papers or exams 41
14. Bibliography 41

Section I. LOGIC OF PROPOSITIONS 43

Chap. 1. SINTAX AND SEMANTICS OF THE LOGIC OF PROPOSITIONS 45

1. Proposition and Propositional Sign 45
2. Language of Propositional Logic 46
3. Logical Operations and Truth Functors. Truth Values 49

4. Logical Operations and Natural Language 52

Chap 2. THE THEORY OF TRUTH FUNCTIONS 57

1. Formulae and Truth functions. The Table of Bivalent Functors 57
2. Truth Tables and Set Theory 59
3. The Principle of Duality 63
4. Truth Function Algebra 65
5. Interdefinibinability and Reduction of logical Operations 66
6. Implicants and Consequences 68
7. Logical Square and the Table of Bivalent bivalent Functors 70

Chap. 3. DECISION METHODS IN PROPOSITIONAL LOGIC 74

1. Satısfiability and validity 74
2. Semantic Methods 75
   2.1. Tipology of Decision Methods 75
   2.2. Matriceal Method 76
   2.3.. Quine's Method 78
   2.4. Method Semantic Tableaux 79
   2.5. Decision Tree Method 83

Chap. 4. SEQUENTIAL CALCULUS 105

1. Logical Consequences and Sequential Calculus 105
2. Sequential Calculus Rules for Propositional Logic 107

Chap. 5. NORMAL FORMS. DECISION BY NORMAL FORMS 115

1. Propositional Equivalences and Normalisation 116
2. Conjunctive Normal Forms 118
3. Disjunctive Normal Forms 120
4. Perfect Disjunctive Normal Forms 122
5. Perfect Conjunctive Normal Forms 125
6. Normal Forms, Models and Systemic Approach 127

Chap. 6. RESOLUTION METHOD FOR PROPOSITIONAL LOGIC 131

1. Classical Inference Schemata and the Principle of Resolution 131
2. Theorem of Resolvent 135
3. Resolutive Derivation and Resolutive Rejection 139
4. Logical Consequence and Resolutive Derivation 143
5. The Principle and the Method of Resolution 145
6. Resolutive Strategies 147

7. Teleologic Resolution 157
8. Closed Tree and Resolutive Rejections 163
9. Dual Resolution, Devision and Minimization 166
9.1. Dual Resolvent and Dual Resolution. Species of dual Resolutions 166
9.2. Principle of Dual Resolution 169
9.3. Dual Resolutive Derivation 171
9.4. The Theorem of Dual Resolutive Derivation 173
9.5. A New Interpretation of the Principle of Dual Resolution 174
9.6. Reduced Formulas, Factor Clause and Minimizations 179
9.7. Sy metric Dual Resolvents 185
9.8. Sy metric Dual Resolution and Minimization 187
9.9 Duality of Robinson's and Maslov's Resolutions 189

Chap. 7. METHOD OF CONSERVATIVE EXTENTION OF PARTIAL INTERPRETATIONS 199

Chap. 8. DAVIS–PUTNAM METHOD 211

1. Rules of Davis–Putnam's Method 211
2. Splitting Rule Theorem 212
3. Splitting Rule Theorem and Resolution Principle 221
4. Perfect Disjunctive Normal Forms and Dual Davis–Putnam Method 227
5. Dual Davis–Putnam Theorem 230
6. Dual Davis–Putnam Method 231

Chap. 9. NORMAL FORMS, RESOLUTION METHOD AND LOGICAL EQUATIONS 237

1. Normal Forms and Problem Solving 237
2. Disjunctive Clauses, And –Or trees and Automated Logical Proof 243
3. Propositional Logic and Logic Programming 248

Section II. PREDICATE LOGIC AND KNOWLEDGE BASIS 253

Cap.1. SINTAX AND SEMANTICS OF PREDICATE LOGIC 255

1. Questions and Problems 255
2. Language of Predicate logic 258
3. Semantic of Predicate Logic 264
4. Predicates and Propositions. Questions and Answers 267

Cap. 2. DECISION METHODS FOR PREDICATE LOGIC 272

1. Semantic Tablaux Method 272

2. Decision Trees Method 276
3. Sequential Calculus for Predicate Logic 288
4. Resolution Method 297
5. Resolutive Strategies for Predicate Logic Formulas 342
5.1. Linear Resolution 344
5.2. Semantic Resolution Strategies 350
Conclusions 355
Selective Bibliography 357

ANEXES – SUBJECTS FOR VERIFICATIONS AND EXAMS 359

Anexa A. Theoretical Subjects for Propositional Logic 359
Anexa B. Propositional Logic Exercises 361
Anexa C. Formalization Exercises. Primitive Predicates and Definitions of Derived Predicates and Propositions 363
Anexa D. Converse Procedure. Decodifications 368
Anexa E. Syllogistic and Predicate Logic Problems 376

INTRODUCERE. OBIECTUL LOGICII

Termenul „logică" desemnează astăzi o mulţime de teorii şi sisteme formale, nu neapărat reciproc consistente, fiecare dintre ele descriind o latură sau un aspect al normelor şi principiilor după care se călăuzesc gândirea corectă şi activităţile eficiente. Toate sistemele logice definesc, într-o formă sau alta, noţiunea de propoziţie adevărată şi regulile după care se trece de la un set de axiome sau premise adevărate la alte propoziţii adevărate.

Relaţiile de inferenţă şi cele de consecinţă logică au stat în centrul preocupărilor logicii din toate timpurile. Din acest punct de vedere, majoritatea teoriilor şi sistemelor logice tind spre definirea unor reguli şi instrumente cu ajutorul cărora să putem garanta conservarea adevărului în trecerea de la axiome la teoreme, de la premise la consecinţe. Pe acest temei, mulţi logicieni definesc astăzi *logica ca o teorie a inferenţelor valide* sau ca *o mulţime de propoziţii închisă faţă de relaţia de consecinţă logică.*

Logica este, în acelaşi timp, o *disciplină clasică* cu o istorie de peste două milenii şi o *disciplină modernă* implicată în dezvoltarea unor ramuri tehnice de vârf, cum sunt ştiinţa computerelor şi inteligenţa artificială, programele expert etc. Ea este, totodată, legată de teoria fundamentelor matematicii şi de metodologia ştiinţelor experimentale. Mai recent, a câştigat teren în analiza unor probleme centrale ale ştiinţelor sociale, cum sunt teoria normelor sau logica deontică, praxiologia formală sau logica acţiunii, teoria organizării şi conducerii, cercetarea deciziei, logica preferinţelor etc.

Pentru a înţelege unitatea şi diversitatea în cercetarea logică modernă este util să menţionăm aici câteva faţete sau dimensiuni ale teoriilor, sistemelor şi metodelor ce caracterizează, definesc şi delimitează ştiinţa logicii astăzi.

1. Teoria logică. Limbajul unei teorii logice

O teorie logică se structurează la mai multe niveluri şi are mai multe dimensiuni sau faţete. Pentru a defini o teorie logică trebuie să-i caracterizăm limbajul, valorile de adevăr, domeniul de interpretare, funcţia de valorizare şi noţiunea de model sau interpretare.

1.1. **Limbajul.** Orice teorie logică modernă se caracterizează printr-un *alfabet* sau *vocabular* şi un set de reguli ce împreună definesc mulţimea formulelor bine formate în limbajul teoriei. Trecerea de la alfabet la limbajul teoriei se face prin intermediul regulilor de formare. În mod obişnuit, alfabetul unei teorii logice conţine mai multe tipuri de semne. În mod necesar, alfabetul trebuie să permită definirea „atomilor" sau a propoziţiilor elementare şi a conectivelor sau operaţiilor logice cu ajutorul cărora se formează propoziţiile compuse. Convenim să notăm în cele ce urmează alfabetul unor teorii logice prin *A* iar limbajul acesteia prin *L*.

1.2. **Valori de adevăr.** O teorie logică admite cel puţin două valori de adevăr, adevărul şi falsul. Logica nu se ocupă de definirea filosofică a adevărului şi falsului. Teoria adevărului ţine de domeniul teoriei cunoaşterii, un capitol important al oricărui sistem complet de filosofie.

În unele teorii logice, ca în logica bivalentă clasică, falsul este definit în raport cu adevărul: tot ceea ce nu este determinat ca adevărat este fals, fiind un concept derivat. În altele, cele două valori sunt acceptate ca două concepte primitive, la fel de îndreptăţite.

Sunt însă sisteme şi teorii logice care admit în afară de adevăr şi fals şi alte valori de adevăr, cum ar fi „necunoscutul" sau „nedeterminatul". Altele disting între adevărul necesar şi cel contingent, falsul contingent şi cel necesar sau admit ca valorizări: adevărul, absenţa oricăror valori, ambele valori, adevărul şi falsul. Tot o partiţie în patru este şi cea care admite: adevărul tare, adevărul, non-falsul şi falsul. Există sisteme logice cu infinit de multe valori cuprinse între 0 şi 1.

Din punctul de vedere al numărului de valori admise, sistemele logice se împart în sisteme bivalente, trivalente, tetravalente, pentavalente, n-valente. Primele logici polivalente au fost dezvoltate de J. Lukasiewicz şi E. Post. Lukasiewicz a propus logici trivalente şi tetravalente. S.C. Kleene a propus o logică trivalentă; la fel logicianul rus Bochvar. La noi, Gr. C. Moisil a dezvoltat mai multe sisteme de logică polivalentă.

Convenim să notăm în cele ce urmează domeniul de valori al unei teorii logice prin

$$V = \{V_1,...,V_n\}$$

1.3. **Domeniul de interpretare.** Unele teorii logice se deosebesc prin domeniul de interpretare, obiectele primitive ale acestuia, relaţiile definite şi proprietăţile lor formale.

O problemă centrală devine, în acest caz, cea a domeniului de referinţă al ştiinţei logicii: priveşte acesta lumea fizică reală, şirurile de semne grafice sau sonore, sensul acestora sesizat de indivizii umani sau înţelesul intersubiectiv instituit istoriceşte într-o limbă naturală, teorie ştiinţifică sau limbaj artificial ? Este logica o ştiinţă despre existenţa naturală şi socială, despre regulile sintactice de construcţie a limbajelor formale, despre înţelesul expresiilor lingvistice sau despre operaţiile logice prin intermediul cărora se prelucrează conţinutul informaţional ataşat semnelor lingvistice ?

Diferitele teorii logice propun opţiuni diferite despre domeniul de interpretare, tipologia obiectelor incluse în acesta şi relaţiile de alternativitate, trecere sau tranziţie între stările sau lumile posibile asociate. Sistemele de logici modale, temporale, dinamice, epistemice, axiologice, deontice, teleologice reclamă domenii de interpretare specifice dotate cu relaţii de alternativitate ce satisfac anumite proprietăţi formale. Convenim să notăm, în continuare, domeniul de interpretare prin W, iar relaţia de alternativitate, definită pe $W \times W$, prin R. Foarte adesea domeniul de interpretare se notează prin D.

1.4. **Funcţia de valorizare.** Admiterea, pur şi simplu, a unui alfabet şi implicit a unui limbaj, precum şi a unui domeniu de valori, nu constituie încă o teorie logică, dacă mulţimea propoziţiilor elementare (sau „atomi") nu e legată printr-o *funcţie de valorizare* cu domeniul de valori admis şi nu se dau reguli de extindere a atribuirilor de valori de la propoziţiile elementare la cele compuse. Aceasta conduce la posibilitatea determinării, pe criterii pur formale, a valorilor de adevăr ale propoziţiilor compuse. Logicile matematice clasice admit funcţiile de adevăr sau exigenţele verifuncţionalităţii. Logicile modale nu mai satisfac această condiţie, valoarea de adevăr a unei formule modale fiind definită, între altele, cu ajutorul structurilor model şi a relaţiilor de alternativitate definite între stările sau lumile posibile. Convenim să notăm funcţia de valorizare sau atribuire de

valori definită pe alfabetul A sau produsul $A \times W$ cu valori în V prin φ, $\varphi : A \rightarrow V$ sau $\varphi : A \times W \rightarrow V$.

1.5. **Propoziţie adevărată, model şi validitate.** În orice teorie logică se definesc pe baza noţiunilor de atom, sau propoziţie elementară, funcţiile de valorizare, domeniul de interpretare, mulţimea valorilor de adevăr, conceptul de propoziţie adevărată şi regulile de extindere a noţiunii de adevăr la diferite tipuri de formule desemnând propoziţii compuse cu ajutorul operaţiilor logice (conjuncţie, disjuncţie, cuantificare, modalizare etc.).

Introducerea conceptului de model sau propoziţie adevărată într-o stare sau lume posibilă printr-o funcţie de valorizare dată permite definirea, apoi, a conceptelor de realizabilitate, validitate, irealizabilitate sau inconsistenţă, invaliditate sau infirmabilitate precum şi a noţiunii fundamentale de consecinţă logică.

Vom reda prin $w \models_{\varphi} A$ enunţul „propoziţia A este adevărată în starea w prin atribuirea de valori φ"

1.6. **Definiţia teoriei logice.** Pe temeiul celor lămurite mai sus (1.1 - 1.5), putem caracteriza, în mod abstract, teoria logică ca un sistem:

$$TL = \{ X, A, L, V, W, R,, \Psi, \varphi \models \}$$

unde:

X reprezintă o mulţime de agenţi sau subiecţi cunoscători, ce admit sau asertează un enunţ, A este un alfabet, L un limbaj, V o mulţime de valori de adevăr, $|V| \geq 2$, W un domeniu de interpretare; $R \subset W \times W$ este o relaţie de alternativitate sau acces dintr-o lume într-alta; Ψ este o funcţie de interpretare definită pe A_T ($A_T \subset A$) cu valori în V, φ este funcţia de asignare de valori variabilelor individuale, iar $\models$ este simbolul asertării veridice. $W \models_{\varphi} A$, se citeşte „propoziţia A este adevărată în starea sau domeniul W prin asignarea φ".

Subsistemul $\langle A, L \rangle$ reprezintă componenta semiotică formală a unei teorii logice. Ea poate fi descrisă cu ajutorul conceptelor lingvisticii matematice sau teoriei sistemelor axiomatice formale şi poartă, de regulă, denumirea de *sintaxă logică*. La acest nivel, o teorie logică este un sistem formal productiv, în care din anumite expresii iniţiale privilegiate sau selectate sunt generate alte expresii selectate sau cuvenite „terminale", cu ajutorul unor reguli. Dar termenul de sintaxă include şi operaţiile calculatorii ce descriu actele de inferenţă validă, care au loc între şiruri sau mulţimi de propoziţii bine formate privilegiate ce pot fi adevărate sau false.

Subsistemul ⟨ A, L, V, Ψ, φ W, R ⟩ înlesneşte descrierea relaţiilor semantice. Ψ este funcţia de valorizare iniţială a propoziţiilor elementare sau atomilor propoziţionali, în timp ce ⊨ descrie, după cum am văzut, o relaţie semantică ternară de forma „propoziţia elementară *A* este adevărată în lumea w prin atribuirea φ" cu ajutorul căreia poate fi definit adevărul agregatelor propoziţionale mai complexe, pe baza principiului verifuncţionalităţii (i.e al dependenţei valorii de adevăr a propoziţiilor complexe de valoarea de adevăr a propoziţiilor elementare) şi celelalte concepte semantice menţionate mai sus (realizabilitate, contradicţie, validitate, etc).

Subsistemele în care intervin şi elemente din *X*, mulţimea subiecţilor cunoscători sau agenţilor acţiunii, se vor numi sisteme logico-pragmatice De regulă, logicile epistemice, deontice, preferenţiale sau axiologice fac uz în modelele şi interpretările lor semantice şi de agenţi, devenind prin aceasta teorii semantico-pragmatice

Din această perspectivă, *logica ne apare ca o teorie a operaţiilor logice sintactice, semantice şi pragmatice care au loc în limbile naturale sau artificiale, operaţii în care se conservă fie denotatul sau semnificaţia expresiilor (în definiţii, diviziuni, clasificări etc) sau valoarea de adevăr „adevărat" (în demonstraţii şi în deducţii)* Relaţia de consecinţă logică, esenţială în orice teorie logică, se defineşte prin proprietatea de a conserva în concluzii sau consecinţe adevărul premiselor sau axiomelor De aceea, se mai spune că o *teorie logic deductivă este o mulţime de propoziţii adevărate, închisă faţă de relaţia de consecinţă logică*

Dar logica nu trebuie redusă la actele sau operaţiile de deducţie sau inferenţă. Argumentarea este şi ea o preocupare majoră a cercetării logice şi nu este reductibilă la teoria actelor inferenţiale. La fel, definiţiile, clasificările, diviziunile, analiza şi sinteza, raţionamentul inductiv, raţionamentul prin analogie sunt şi ele operaţii studiate de logica modernă, fără a putea fi subsumate simplist teoriei inferenţelor.

2. Principiile logice şi tipologia sistemelor logice

Două teorii logice sau două sisteme logice se pot deosebi în privinţa: 1) semnelor primitive admise sau alfabetului; 2) regulilor de formare; 3) numărului şi naturii valorilor de adevăr; 4) domeniului de interpretare şi structurilor de model admise; 5) modului de extindere a valorilor de adevăr de la propoziţiile elementare la cele complexe; 6) schemelor de inferenţă sau modului în care este definită relaţia de consecinţă logică; 7) principiilor postulate.

Criteriile menţionate permit clasificarea mono sau multicriterială a teoriilor logice.

Două teorii având la bază acelaşi alfabet se pot deosebi prin regulile de formare sau prin regulile de inferenţă, prin axiomele sau principiile pe care le postulează. Aceste deosebiri dintre teoriile logice vor ieşi la iveală de îndată ce vom trece la prezentarea logicii propoziţiilor, logicii predicatelor, a logicilor modale şi polivalente, a logicii intuiţioniste etc. În cele ce urmează vom pune în lumină câteva principii logice care diferenţiază tipologia teoriilor logice, în primul rând din punct de vedere al numărului şi naturii valorilor şi al modului de definire a funcţiei de valorizare.

2.1. **Valenţa sistemelor logice.** Orice teorie logică are un număr determinat de valori de adevăr distincte ($2 \leq n < \infty$). Aceasta înseamnă că pentru orice teorie logică putem determina un cardinal al lui V, $|V|$, constând în numărul valorilor de adevăr pe care le admite şi putem enumera explicit aceste valori, $V = \{ v_1,...,v_n \}$. Putem formula principiul n-valenţei astfel:

$$|V| = n \qquad (2.1.1.\ \alpha)$$

$$V = [\ v_1,...,v_n\ ] \qquad (2.1.1.\ \beta)$$

Cele mai răspândite teorii logice sunt cele bivalente, utilizând doar două valori, adevărul şi falsul. Principiul bivalenţei poate fi formulat astfel:

$$|V| = 2 \qquad (2.1.1.\ a)$$

$$V = \{ a, f \} \qquad (2.1.1.\ b)$$

Sistemele logice ce respectă principiul bivalenţei au fost numite de Gr. C. Moisil *sisteme chrisipiene.* Logica aristotelică, logica stoică, logica propoziţiilor, logica predicatelor, logicile modale şi temporale sunt, în genere, sisteme logice bivalente. Dimpotrivă, sistemele create de Post şi Lukasiewicz, de S. C. Kleene şi Gr. C. Moisil, D. Bochvar, J. H. Dunn, G. Epstein şi alţii nu sunt sisteme bivalente.

Condiţia 2.1.1.a stabileşte numărul valorilor, iar 2.1.1.b precizează care anume sunt acestea: a (adevărul) şi f (falsul). Putem, explorativ, restrânge ideea de bivalenţă la prima condiţie şi anume, de exemplu, $V = \{ a^N, f^N \}$ sau $V = \{f^C, a^C\}$ (unde a^N, a^C, f^N, f^C desemnează valorile adevăr necesar, adevăr contingent, fals necesar şi fals contingent).

Dacă vom admite

$$|V| = 3 \qquad (2.1.2\ a)$$

$$|V| = \{ a, f, u \} \qquad (2.1.2\ b)$$

şi definiţiile pentru $\neg$ şi $\wedge$:

p	$\neg$p	p$\wedge$q	a	f	u
a	f	a	a	f	u
f	a	f	f	f	f
u	u	u	u	f	u

Tabelul 2 1 2 c Definiţia negaţiei şi a conjuncţiei în sistemul trivalent al lui S.C. Kleene

obţinem sistemul trivalent tare al lui Kleene. Prin *u* s-a notat valoarea „necunoscut"

Dacă vom păstra 2.1.2 a şi schimbăm 2.1.2 b, în 2.1.2 d

$$V = \{ a, f, i \} \qquad (2.12\ d)$$

acceptând, totodată, pentru $\neg$ şi $\supset$ definiţiile:

p	$\neg$p	p$\supset$q	a	f	ı
a	f	a	a	f	ı
f	a	f	a	a	a
ı	i	i	ı	ı	a

(2.12 e)

Tabelul 2 1 2 e Logica trivalentă a lui Lukasiewicz

obţinem logica trivalentă a lui J. Lukasiewicz, creată pentru descrierea viitorilor contingenţi. Prin *i* s-a notat starea „nevalorizabilă" în prezent a unei propoziţii despre un eveniment viitor şi nu simpla necunoaştere de către subiect a valorii ei de adevăr.

În sfârşit, dacă vom schimba 2.1.2. b în 2.2.2. f:

$$V = \{ a, f, m \} \qquad (2.1.2.\ f)$$

cu m înţeles ca „absurd", vom admite definiţiile lui $\neg$ şi $\supset$ ca:

p	⌉p	p⊃q	a	f	m
a	f	a	a	f	m
f	a	f	a	a	m
m	m	m	m	m	m

Tabloul 2.1.2.g. Sistemul trivalent al lui Bochvar

şi un operator de asertare notat prin T

p	Tp
a	a
f	f
m	f

obţinem sistemul trivalent al lui Bochvar creat pentru explicarea unor paradoxuri semantice şi având o interpretare computaţională convenabilă.

Din exemplele de mai sus rezultă că numărul şi natura valorilor, ca şi modul de definire a operaţiilor logice reprezintă un criteriu important în caracterizarea unei teorii logice. Din acest punct de vedere reţinem, deci, existenţa unor teorii logice bivalente, trivalente, tetravalente, n-valente etc.

2.2. **Principiul consistenţei sau noncontradicţiei.**

Dacă admitem că unei propoziţii putem să-i acordăm într-un domeniu dat *W* şi printr-o operaţie de valorizare φ o singură valoare de adevăr din V, atunci vom spune că valorizările noastre sunt consistente sau noncontradictorii. Vom numi această cerinţă drept *principiul consistenţei valorizărilor*. Fie p o propoziţie elementară din *A* şi *W* domeniul de interpretare sau referinţă a lui p. Atunci funcţia φ asociază în *W* fiecărei propoziţii elementare p din *A* două submulţimi $W_1 = \{ w \in W \quad \varphi(p) = a \}$ şi $W_2 = \{ w \in W \quad \varphi(p) = f \}$ Convenim să notăm în continuare pe W_1 prin $\varphi_a(p)$ şi pe W_2 prin $\varphi_f(p)$. $\varphi_a(p)$ va desemna, deci, cazurile în care p este adevărată în domeniul *W*, iar φ_f (p), cazurile în care p este falsă.

φ_a (p) mai este numită de unii autori drept funcţie caracteristică a propoziţiei p în domeniul de interpretare *W*. Principiul consistenţei va putea fi astfel formulat pentru cazul logicilor bivalente:

I $\qquad \varphi_a(p) \cap \varphi_f(p) = \varnothing \qquad (2.2.1)$

2.2.1 postulează separarea sau disjuncţia cazurilor de adevăr şi fals ale unei propoziţii. Principiul consistenţei poate fi generalizat pentru cazul logicilor n-valente, cu

$$V = \{ v_1,...,v_n \}.$$

$$\varphi v_i (p) \cap \varphi v_j (p) = \varnothing \qquad (2.2.2)$$

unde $1 \leq i, j \leq n$ şi $i \neq j$.

Dacă v_i şi v_j sunt două valori de adevăr distincte din V, atunci mulţimea cazurilor în care propoziţia p ia valoarea v_i este disjunctă de mulţimea cazurilor în care p ia valoarea v_j. Prin principiul consistenţei se interzice, deci, ca o propoziţie să ia simultan două valori de adevăr distincte.

Putem, evident, concepe şi sisteme logice în care principiul consistenţei să fie încălcat.

$$\varphi v_i (p) \cap \varphi v_j (p) \neq \varnothing \qquad (2.2.3)$$

respectiv una şi aceeaşi propoziţie să primească prin φ două sau mai multe valori de adevăr. În acest caz, funcţia de valorizare va fi definită pe $A \times 2^W$, respectiv pe produsul cartezian al alfabetului cu mulţimea submulţimilor lui W cu valori în V. Astfel de logici au fost numite *suprasaturate* sau *paraconsistente*. Un model sau interpretare convenabilă pentru o astfel de teorie este admiterea că o propoziţie elementară din A poate fi evaluată de un subiect cunoscător într-un interval de timp redat într-o secvenţă de lumi posibile ca având un şir de valori şi nu o valoare unică. La Congresul de Logică, Metodologie şi Filosofia Ştiinţei de la Moscova (august 1987) s-au purtat vii discuţii despre sistemele de logică paraconsistente.

2.3. **Principiul excluderii celei de a *n + 1* valori.**

Fie A o mulţime de propoziţii elementare sau atomi, $V = \{ v_1,...,v_n \}$ mulţimea de valori de adevăr, W domeniul de interpretare şi φ funcţia de valorizare definite pe produsul $A \times W$ cu valori în V Notând ca mai sus, prin $\varphi v_i(p)$ mulţimea situaţiilor sau lumilor posibile în care propoziţia p ia valoarea v_i, putem descrie asamblıst *principiul excluderii celei de a* n + 1 *valorizări* prin egalitatea:

$$\varphi_{v1}(p) \cup \varphi_{v2}(p) \cup \ldots \cup \varphi_{vn}(p) = W \qquad (2.3.1)$$

Dacă acceptăm principiul bivalenţei, respectiv $|V| = 2$ şi $V = \{a, f,\}$ atunci 2.3.1 devine *principiul terţului exclus*:

$$\text{II} \quad \varphi_a(p) \cup \varphi_f(p) = W \qquad (2.3.2)$$

în formularea sa semantic asamblistă.

Principiul excluderii celei de a $n + 1$ valorizări (2.3.1) afirmă că nu există în domeniul W nici un obiect, caz, lume posibilă sau stare despre care să nu putem spune că propoziţia p ia o valoare de adevăr din domeniul V. Altfel spus, reuniunea cazurilor în care propoziţia p ia valoarea v_1 cu cele în care p ia valoarea v_2, v_3 şi aşa mai departe, până la v_n, epuizează domeniul de referinţă W.

Principiul terţului exclus afirmă că domeniul de interpretare W este epuizat de reuniunea cazurilor în care o propoziţie oarecare p este adevărată cu cele în care aceasta este falsă (2.3.2). Principiul terţului exclus nu exclude ca una şi aceiaşi propoziţii să ia două valori de adevăr – aceasta o interzice principiul consistenţei sau noncontradicţiei, el cere doar valori acceptate şi previne posibilitatea atribuirii unei a treia valori diferită de cele două admise.

Principiul terţului exclus interzice şi nevalorizarea vreunei propoziţii din mulţimea A. Cu alte cuvinte, el preîntâmpină cazul când funcţia de valorizare φ ar fi doar parţial definită pe mulţimea A. Nu există, aşadar, propoziţii pe care să nu le putem eticheta ca adevărate sau ca false.

În logica propoziţiilor *principiul terţiului exclus* este exprimat cu ajutorul operaţiilor de negaţie şi disjuncţie inclusivă.

$$p \vee -p \qquad (2.3.3.)$$

ştiind că negaţia este operaţia logică ce schimbă valoarea de adevăr a unei propoziţii cu valoarea complementară, putem scrie:

$$\varphi_a(-p) = \varphi_f(p) \qquad (2.3.4.)$$

$$\varphi_f(-p) = \varphi_a(p) \qquad (2.3.5)$$

de unde prin regula substituirii echivalentelor în 2.3.2 obţinem:

$$\varphi_a(p) \cup \varphi_a(-p) = W$$

ceea ce reproduce semantic asamblist pe 2.3.3.

Încălcarea principiului terţului exclus şi, corespunzător, a principiului excluderii celui de al n + 1-lea caz, poate fi descrisă formal prin :

$$\varphi_a(p) \cup \varphi_f(p) \subset W \qquad (2.3.6)$$

$$\varphi_{v1}(p) \cup \varphi_{v2}(p) \cup \ldots \cup \varphi_{vn}(p) \subset W \qquad (2.3.7)$$

Semnul includerii stricte joacă un rol esenţial în (2.3.6) şi (2.3.7). Putem, evident, rescrie cazurile de încălcare a principiilor de mai sus şi cu ajutorul semnului „*" pentru adăugarea celei de a $n + 1$ valorizări.

$$\varphi_a(p) \cup \varphi_f(p) \cup \varphi^*(p) = W \qquad (2.3.8)$$

$$\varphi_{v1}(p) \cup \quad \cup \varphi_{vn}(p) \cup \varphi^*(p) = W \qquad (2.3.9)$$

Prin $\varphi^*(p)$ am notat cea de-a $n + 1$ valorizare.

În aceste cazuri φ va fi o funcţie parţial definită.

Să restrângem, pentru moment, analiza noastră la logicile bivalente ce utilizează ca valori doar adevărul şi falsul şi să cercetăm cum pot fi clasificate teoriile logice din punctul de vedere al principiilor consistenţei (2.2.1) prin I şi principiul terţului exclus (2.3.2) prin II.

Putem distinge, după E. D. Smirnova, patru alternative; I, II; I, ¬ II; ¬ I, II; ¬ I, ¬ II, respectiv admiterea ambelor principii, ceea ce conduce la logica matematică clasică, admiterea primului principiu şi respingerea celui de al doilea, ceea ce duce la logica cu valorizări lacunare de tipul logicilor trivalente produse de S. C. Kleene, J. Lukasiewicz şi D. A. Bochvar sau la logici intuitiviste de tipul celor propuse de A. Heyting; respingerea primului principiu şi admiterea celui de al doilea, ceea ce duce la logici cu valorizări suprasaturate de tipul logicilor paraconsistente, în care o propoziţie poate primi simultan mai multe valori şi, în sfârşit, respingerea ambelor principii, ceea ce duce la logici cu semantici aproximative sau la aşa-numitele logici relevante, în care unele propoziţii pot primi mai multe valori, iar altele pot rămâne nevalorizate.

2.4. **Principiul constanţei în valorizarea unei propoziţii.** Într-o teorie logică coerentă acordarea unei valori, unei propoziţii elementare trebuie să rămână aceiaşi în toate apariţiile semnului propoziţional, indiferent de numărul acestora, atât timp cît luăm în considerare interpretarea sau modelul respectiv. Este cerinţa de constanţă şi consecvenţă în interpretare care poate fi corelată şi cu *principiul identităţii* din logica tradiţională.

$$p \rightarrow p \qquad \text{(PI)}$$

Atribuirea aceleiaşi propoziţii a unor valori deosebite ne scoate din perimetrul logicilor clasice, trecându-ne în teorii logice nemonotone, paraconsistente, relevante etc.

2.5. **Principiul verifuncţionalităţii.** Sistemele logice clasice presupun dependenţa valorii de adevăr a propoziţiilor compuse de valoarea de adevăr a propoziţiilor elementare. De aceea, teoria funcţiilor de adevăr reprezintă un capitol esenţial în logica propoziţiilor.

Sunt posibile două demersuri: de la cunoaşterea valorii de adevăr a propoziţiilor elementare să conchidem asupra cunoaşterii valorii de adevăr a propoziţiilor compuse sau invers, punând anumite ipoteze asupra valorii de adevăr a propoziţiilor compuse să inferăm, pe baza unor reguli de inferenţă, asupra valorii de adevăr a propoziţiilor elementare şi să găsim un model sau un contramodel şi în cele din urmă să conchidem cu certitudine asupra valorii de adevăr a propoziţiilor compuse.

După modul cum se raportează la elementele sistemului *TL* (vezi 1.5), la principiile logice ale bivalenţei, consistenţei, terţiului exclus, constanţei în valorizare şi verifuncţionalităţii, vom distinge diferite tipuri de teorii şi sisteme logice logice. Astfel, o primă distincţie o vom face între logicile bivalente şi polivalente (trivalente, tetravalente, n-valente). De asemenea, vom deosebi în funcţie de admiterea principiilor consistenţei şi terţiului exclus între logicile clasice şi logicile deviante sau neclasice care încalcă primul sau al doilea dintre principiile menţionate. *Logicile deviante* schimbă unele teze sau principii ale logicii clasice. Sunt în schimb, teorii logice neclasice care se dezvoltă pe temelia sistemelor logice matematice clasice, adăugând acestora noi operatori de factură modal-aletică, epistemică, deontică, temporală, teleologică, preferenţială. Astfel de teorii logice vor fi sisteme modale aletice T, S4, S5, B, S4.2, S4.3, D sau sistemele de logică temporală P4, P5.

Putem clasifica sistemele logice şi după numărul argumentelor operatorilor modali în sisteme modale monadice, diadice, ternare. Astfel, sistemele T, S4, S5, etc. sunt monadice, în timp ce sistemul lui von Wright din „A new System of Deontic Logic" (1971) este un sistem diadic. Diadice sunt sistemele de logica preferinţelor, cele de logică epistemică, teleologică etc.

Un alt criteriu de clasificare a teoriilor logice îl reprezintă prezenţa sau absenţa dimensiunii pragmatice în teoria logică în cauză. De regulă, logicile preferenţiale, epistemice sau teleologice fac explicită prezenţa agenţilor sau subiecţilor cunoscători.

O clasificare a sistemelor logice se mai poate face după criteriul structurii de modele utilizate şi după proprietăţile relaţiei de alternativitate sau după tipologia semanticilor utilizate, referenţiale, de lumi posibile, procedurale, topologice etc.

Frecvent, teoriile şi sistemele logice sunt clasificate şi după proprietăţile algebrice pe care le au mulţimile lor de formule bine formate sau tautologiile sau formulele identic-adevărate. Astfel, sistemelor logice polivalente, modale, temporale le vor fi asociate diferite structuri algebrice.

3. **Consecinţe logice**

Teoriile logice descriu exigenţe ale dinamicii gândirii umane corecte, actele de trecere de la premise sau date iniţiale la cunoştinţe derivate, respectiv la concluzii şi consecinţe. Noţiunea de consecinţă logică este centrală în orice sistem sau teorie logică. Definirea ei depinde de natura limbajului logic, de valorile de adevăr admise, de tipul de construcţie semantică asociat, de funcţia de interpretare şi de caracterizarea conceptului de propoziţie adevărată. În logica clasică a propoziţiilor şi în logica predicatelor, admiţând principiile consistenţei şi terţiului exclus, vom defini relaţia de consecinţă logică semantică, în cadrul teoriei modelelor. Un model este o funcţie care asociază fiecărei formule bine formate o valoare de adevăr, de regulă, dintr-o logică bivalentă, unde $V = \{0, 1\}$.

Vom introduce, mai întâi, noţiunile de realizabilitate, infirmabilitate, validitate şi contradicţie şi, în sfârşit, noţiunea de consecinţă logică, arătând, între altele, variaţia acesteia în funcţie de atitudinea adoptată de creatorul sistemului faţă de principiile consistenţei şi terţiului exclus.

Fie A_1 o mulţime de propoziţii elementare (variabile propoziţionale), L limbajul logicii propoziţiilor, B şi C propoziţii. Notăm, de asemenea, prin W domeniul de interpretare, prin $V = V_2 = \{a, f\}$ mulţimea valorilor de adevăr şi prin φ funcţia de valorizare: $\varphi: A_1 \times W \rightarrow V_2$. Prin $w_\varphi \models B$ vom prescurta enunţul semantic aparţinând metalibajului, „Propoziţia B este adevărată în starea w prin valorizarea φ", respectiv, $w_\varphi \models B$ este totuna cu $\varphi(B, w) = a$, care mai poate fi citită: „Propoziţia B are în starea w valoarea de adevăr a" Ca şi anterior, prin $\varphi_a(B)$ vom nota mulţimea stărilor în care

propoziţia B este adevărată. Admitem că $p \in A_1$. Prin $w_\varphi \models B$ vom nota „Propoziţia B este falsă în starea w şi atribuirea de valori φ". Definim, în continuare:

$$w_\varphi \models p \Leftrightarrow w \in \varphi_a(p) \tag{3.1}$$

$$w_\varphi \models\!\!\!| p \Leftrightarrow w \in \varphi_f(p) \tag{3.2}$$

Astfel definite, adevărul şi falsul sunt ambele concepte primare, autonome. Extindem conceptele de adevăr şi fals la propoziţiile complexe, obţinute prin negaţie, conjuncţie, disjuncţie, inclusiv implicaţie.

$$\varphi_a(-B) = \varphi_f(B) \tag{3.3}$$

$$\varphi_f(-B) = \varphi_a(B) \tag{3.4}$$

$$\varphi_a(B \,\&\, C) = \varphi_a(B) \cap \varphi_a(C) \tag{3.5}$$

$$\varphi_a(B \vee C) = \varphi_a(B) \cup \varphi_a(C) \tag{3.6}$$

$$\varphi_a(B \supset C) = \varphi_f(B) \cup \varphi_a(C) \tag{3.7}$$

$$\varphi_f(B \,\&\, C) = \varphi_f(B) \cup \varphi_f(C) \tag{3.8}$$

$$\varphi_f(B \vee C) = \varphi_f(B) \cap \varphi_f(C) \tag{3.9}$$

$$\varphi_f(B \supset C) = \varphi_a(B) \cap \varphi_f(C) \tag{3.10}$$

Conceptele de realizabilitate, infirmabilitate, validitate şi contradicţie se definesc uşor pe baza noţiunilor de adevăr şi fals ($w_\varphi \models B$ şi $w_\varphi \models\!\!\!| B$). Vom nota în ordine aceste concepte prin: *realiz, infirm, valid, irealiz*

$$realiz(B) \Leftrightarrow \exists w \exists \varphi\ (w_\varphi \models B) \tag{3.11}$$

$$infirm(B) \Leftrightarrow \exists w \exists \varphi\ (w_\varphi \models\!\!\!| B) \tag{3.12}$$

$$valid(B) \Leftrightarrow \forall w \forall \varphi\ (w_\varphi \models B) \tag{3.13}$$

$$irealiz(B) \Leftrightarrow \forall w \forall \varphi\ (w_\varphi \models\!\!\!| B) \tag{3.14}$$

O formulă B este realizabilă (infirmabilă) dd, există o stare w şi o valorizare φ astfel că B este adevărată (falsă) în starea w prin valorizarea φ. Analog, o formulă este validă (contradicţie), dd, este adevărată (falsă) în orice stare w şi prin orice valorizare φ.

Putem acum defini mai multe concepte de consecinţă logică semantică:

$$B \models C \Leftrightarrow \forall w \forall \varphi (w \models_{\varphi} B \supset w \models_{\varphi} C) \quad (3.15)$$

$$B_1,...,B_n \models_{\varphi} C \Leftrightarrow \forall w \forall \varphi (w \models_{\varphi} B_1 \& ... \& B_n \supset w \models_{\varphi} C) \quad (3.16)$$

$$valid\,(B) \models_{\varphi} valid\,(C) \Leftrightarrow valid\,(B) \supset valid\,(C) \quad (3.17)$$

Spunem că C este o *consecinţă logică* a lui B dd, orice stare şi orice valorizare care face pe B adevărat face, de asemenea, şi pe C adevărat (3.15). Următoarea consecinţă logică (3.16) este între o mulţime de premise $B_1,...B_n$ şi o concluzie C, iar ultima (3.17) descrie o relaţie de consecinţă semantică între propoziţii valide.

Deocamdată, ceea ce ne interesează este doar relevarea locului central pe care îl are conceptul de consecinţă logică semantică în sistemele logice moderne. Interesul nu este doar teoretic, ci şi practic aplicativ. Demonstratoarele automate de teoreme nu numai că presupun o definiţie corectă a relaţiilor de consecinţă logică, dar presupun, în acelaşi timp, tehnici şi metode de prelucrare logică formală a bazelor de cunoştinţe transpuse în sisteme de semne adecvate.

Conceptul de consecinţă logică este central şi în demonstraţiile matematice şi în raţionamentele din teoriile fizice, din cele economice, juridice sau social-politice.

Consecvenţa în gândire, admiterea numai a unor consecinţe care derivă din premise adevărate, are o mare importanţă şi în activităţile practice, tehnico-inginereşti ca şi în viaţa cotidiană.

Este firesc să ne întrebăm care sunt proprietăţile relaţiei de consecinţă logică semantică şi ce rol joacă acestea în demonstraţia logică?

Relaţia de consecinţă logică este reflexivă, tranzitivă şi monotonă.

P1. Dacă Q aparţine unui set de premise sau ipoteze Γ, atunci din Γ se deduce Q.

Altfel spus, Q este o consecinţă logică din Q şi alte premise. Ipotezele iniţiale trebuie admise şi mai târziu.

Dacă $Q \in \Gamma$, *atunci* $\Gamma \models Q$ (reflexivitate) (3.18)

P2. Dacă din Γ se deduce A şi din Γ şi A se deduce Q,, atunci din Γ se deduce Q.

Dacă $\Gamma \models A$ *şi* $\Gamma, A \models Q$, *atunci* $\Gamma \models Q$ (tranzitivitate) (3.19)

P3. Dacă dintr-un set de premise sau ipoteze se deduce o consecinţă, aceasta se va deduce şi din premisele sau ipotezele iniţiale completate cu un alt set de ipoteze sau premise.

Dacă $\Gamma \models Q$, *atunci* $\Gamma, \Gamma' \models Q$ (monotonicitate) (3.20)

Relaţia de consecinţă logică poate fi particularizată la un anumit model *s* sau la o mulţime de modele ce defineşte o semantică.

O *semantică* S poate fi caracterizată, după Dov Gabbay, ca o mulţime de modele pentru care este definită relaţia de consecinţă logică, $\Delta \models_s \Gamma$, astfel încât să satisfacă condiţiile:

$$(\forall s \in S)\ [\ \forall X \in \Delta (s(X)=1 \supset \exists Y \in \Gamma (\ s(Y) = 1\ )\ ] \qquad (3.21)$$

Relaţia $\Delta \models_s \Gamma$ se citeşte: „Din mulţimea de ipoteze Δ se deduce în modelul *s* mulţimea de consecinţe Γ" iar 3.21 se citeşte: „Pentru orice model *s* din S, dacă sunt adevărate în *s* toate ipotezele din Δ, atunci există în Γ cel puţin o formulă adevărată în modelul *s*"

Relaţia de consecinţă logică a fost introdusă mai sus ca un concept semantic. Este util să relevăm că ea poate fi captată şi cu ajutorul unor proceduri sau sisteme algoritmice.

Relaţia de consecinţă logică poate fi descrisă în cadrul unor sisteme axiomatice sau cu ajutorul unor procedee de decizie, cu ajutorul regulilor calculului natural sau ale calculului secvenţial. Dar despre acestea vom vorbi în capitolele următoare.

4. Logica şi legile gândirii

Nu numai specialiştii în logică, dar şi cei din alte profesii sunt tentaţi să asocieze obiectul logicii cu cercetarea principiilor şi legilor gândirii umane. Logica, scria George Boole în 1854 este chemată să cerceteze, în primul rând, „legile fundamentale ale operaţiilor mentale prin care este înfăptuit raţionamentul"[1] „Ea, scria Boole, se ocupă de două feluri de relaţii, de relaţii dintre lucruri şi de relaţii dintre fapte"[2] Acestea din urmă sunt exprimate prin propoziţii.[2]

De aceea, cel puţin pentru scopurile logicii, relaţiile dintre fapte pot fi reduse la relaţii între propoziţii. Premisele oricărui raţionament logic exprimă relaţii date între anumite elemente şi concluzia trebuie

[1] George Goole, *An Investigation of the Laws of Thought*, Dover Publications, Inc. p. 3

[2] Ibid., p. 3.

să exprime o relaţie implicită între acele elemente. După George Boole, există legi ştiinţifice ale gândirii şi raţionamentului şi acestea au un caracter matematic aparţinând cunoaşterii pozitive, exacte.

Fondatorul logicii matematice, George Boole, nu se sfia să afirme că logica se ocupă de operaţiile intelectului uman, de procese mentale surprinse prin intermediul alcătuirilor propoziţionale. Cei mai de seamă dintre logicienii contemporani admit şi ei că logica se ocupă de analiza propoziţiilor sau a judecăţilor, de cercetarea inferenţelor şi a demonstraţiilor, acordând atenţie formei şi făcând abstracţie de conţinut[3] Teoria inferenţei, actul deducerii a fost o preocupare statornică în studiul logicii. În sens restrâns, logica este, după Patrick Suppes, teoria argumentelor valide sau teoria inferenţelor deductive[4] Într-un sens mai extins ea cuprinde şi teoria definiţiei şi, într-un sens mai larg, ea cuprinde şi teoria mulţimilor şi a sistemelor axiomatice[5]

În opoziţie cu punctul de vedere după care logica este relevantă pentru cercetarea legilor gândirii umane, Jan Lukasiewicz, fondatorul logicilor polivalente, scria: „Totuşi nu este adevărat că logica este ştiinţa despre formele gândirii. A cerceta cum gândim sau cum trebuie să gândim nu aparţine obiectului logicii. Prima cercetare aparţine psihologiei, cea de a doua unei arte practice, mnemonica. Logica nu are mai mult de a face cu gândirea decât matematica"

Jan Lukasiewicz consideră vagă sau imprecisă expresia „formă a gândirii" Gândirea, susţine autorul polonez, este un fenomen psihic, iar fenomenele psihice nu au întindere şi deci, nici forme.

Admitem, ca şi Lukasiewicz, că psihologia studiază, între altele, procesele gândirii la nivelul subiectului uman, că aceasta se interesează de reflectarea senzorială, de percepţii şi de reprezentări şi, în acelaşi timp, de geneza noţiunilor şi de actele de judecată, de procesul de abstractizare, de comunicare şi de înţelegere, de inteligenţă şi de creativitate etc., la nivelul individului uman sau la nivelul unor activităţi umane. Toate acestea fiind studiate, deopotrivă prin observaţie şi experienţă, prin modelare, metode matematice statistice şi probabilistice. Dar de aici nu rezultă, nici un moment, că

---

[3] Alonzo Chuech, *Introduction to Mathematical Logic*, Princeton, 1956, p. 14.

[4] Patrick Suppes, *Introduction to Logic*, D. van Nostrand Company, Princeton 1957, p XVIII

[5] Jan Lukasiewicz, *Aristotel's Syllogistic from Standpoint of Modern Formal Logic*, Oxford, 1957, 48.

logica nu ar avea nimic de spus în privinţa legilor şi metodelor prin care mintea umană trece de la premise la concluzii în actul gândirii deductive sau despre tipologia operaţiilor prin care noi definim sau clasificăm un obiect sau o mulţime de obiecte. Frecvent, două sau mai multe discipline, au acelaşi obiect de referinţă în planul existenţei materiale sau social umane, ele deosebindu-se prin scopul asumat şi metodele utilizate, prin tipul de rezultate furnizate şi prin aplicaţiile practice pe care le propun. Nu rezistă la o analiză mai atentă nici obiecţia lui Jan Lukasiewicz împotriva expresiei „formă a gândirii"

Dacă gândirea este exprimată prin rostiri sau propoziţii, iar acestea sunt succesiuni de sunete sau de semne grafice, ele pot avea şi au anumite forme cu proprietăţi perfect decelabile. În plus, putem distinge, în afară de formele şi structurile spaţiale, forme şi structuri temporale, dinamice. Algoritmii utilizaţi în matematică şi în limbajele de programare sunt astfel de forme sau structuri dinamice, proceduri sau reguli de rezolvare a unor clase de probleme. Expresia „forme ale gândirii", poate fi şi ea dezambiguizată, de îndată ce determinăm un anumit vocabular primitiv şi regulile de formare a expresiilor într-un limbaj dat, de îndată ce definim conceptele de realizabilitate şi validitate, de lege logică într-un anumit calcul logic matematic. În consecinţă, vom admite ca şi George Boole şi numeroşi alţi logicieni contemporani, că *logica pune în lumină legi şi principii ale gândirii corecte, structuri şi reguli de inferenţă validă, tehnici şi metode de definire şi de clasificare, structuri de argumentare, explicaţie, predicţie, întemeiere* etc., utilizând pentru aceasta analiza semnelor şi expresiilor propoziţionale din limbile naturale sau din limbajele teoriilor ştiinţifice.

Atât Lukasiewicz, cât şi B. Russell sau H. B. Curry, atunci când au contestat relevanţa logicii pentru studierea legilor gândirii, au urmărit eliminarea psihologismului, respectiv a tentativelor de a reduce obiectul logicii la psihologie, în spiritul tezelor apărate de John Stuart Mill, C. Sigwart, W. Wundt, B. Erdmann. După cum remarca logicianul român Petre Botezatu, „logica studiază gândirea detaşată de subiect (...) în funcţionarea sa ideală, netulburată de intervenţii străine, aptă de a transmite adevărul din verigă în verigă. Psihologia studiază gândirea ca proces, în timp ce logica examinează gândirea ca operaţie"[6] Logica se interesează de validitatea sau invaliditatea raţionamentelor şi de

[6] Petre Botezatu, *Constituirea logicităţii*, Editura Ştiinţifică şi Enciclopedică, 1983, p. 59.

criteriile de verificare a acesteia, în timp ce psihologia, neurofiziologia sau medicina cercetează caracterul normal sau patologic al actelor de gândire la un individ sau altul, rolul diferitelor structuri neuronale în înfăptuirea actelor perceptive, în recunoaşterea formei obiectelor, în geneza reprezentărilor şi noţiunilor etc.

5. Logica şi legile existenţei

Ideea unei concordanţe între principiile existenţei şi principiile logicii o găsim încă la Aristotel: „Prin aceste principii ale demonstraţiei înţeleg propoziţiile generale de care cineva se slujeşte spre a dovedi ceva, cum, de pildă, axioma că despre orice lucru trebuie neapărat sau să afirmăm sau să negăm ceva şi că este peste putinţă ca un lucru să fie şi să nu fie în acelaşi timp, cât şi alte premise asemănătoare"[7] Astfel, dacă x este un obiect şi P este o proprietate, atunci are loc $(x)(P(x) \vee -P(x))$ şi respectiv $(x) -(P(x) \& -P(x))$.

În viziunea aristotelică, principiile terţiului exclus, al non-contradicţiei, ca şi principiul identităţii ($A \equiv A$) sunt valabile, deopotrivă, la nivel ontic şi la nivel logic, la nivelul lucrurilor şi la nivelul enunţurilor.

Mai mult, se susţine că principiile logicii reflectă principiile existenţei. Ideea s-a menţinut de-a lungul secolelor. După Hegel, ştiinţa logicii constituie metafizica propriu-zisă sau filosofia pură, speculativă. Logica, scrie Hegel, se divide în trei părţi: I. *teoria fiinţei*; II. *teoria esenţei*; III. *teoria conceptului şi a ideii* [8]. Astfel caracterizată, logica devine o disciplină filosofică speculativă cu valenţe ontologice, metodice şi cognitive.

În acelaşi spirit, Lenin susţinea că logica şi teoria cunoaşterii trebuie deduse din „dezvoltarea întregii vieţi, a naturii şi a spiritului" şi că logica nu este doar ştiinţa formelor exterioare ale gândirii, ci ştiinţa formelor de dezvoltare a tuturor lucrurilor materiale, naturale şi spirituale, adică a legilor de dezvoltare a întregului conţinut concret al

[7] Aristotel, *Metafizica*, III (B),996, b.c., Editura Academiei RSR, 1965, p. 105.

[8] G.W. F. Hegel, *Enciclopedia ştiinţelor filozofice*, Logica, Editura Academiei RPR, 1962, p. 107.

lumii şi a cunoaşterii acesteia: adică bilanţul, suma, concluzia istoriei cunoaşterii lumii"[9].

Pentru Hegel logica e cercetarea principiilor devenirii existenţei fizice, sociale şi spirituale. Logica devine astfel o teorie generală a întemeierii şi raţionalităţii în orice domeniu al realităţii. Astfel înţeleasă, logica devine un proiect sau o doctrină filosofică şi nu o disciplină matematică, operaţională, constructivă şi controlabilă, aptă de analiză formală şi de aplicaţii tehnice. Moris R. Cohen şi Ernst Nagel considerau că logica poate fi definită ca studiul caracterelor celor mai generale, celor mai dominante a tot ceea ce există şi a tot ceea ce ar putea exista. De aceea principiile logicii sunt formale[10] În mod analog, Gonseth definea logica ca studiul obiectului oarecare. Evident, în limitele lucrării de faţă opţiunea noastră tinde spre prezentarea logicii ca o suită de teorii despre principiile şi legile prelucrării conţinutului informaţional veridic stocat în semnele şi expresiile propoziţionale.

Prezentarea legilor logicii ca o reproducere a legilor existenţei reprezintă o viziune liniar-mecanicistă. Legile logicii nu emană spontan din legile existenţei, chiar dacă acestea din urmă sunt descrise, de regulă, prin enunţuri universale. Nu avem un analog logic al fiecărei legi fizice, al legilor mecanicii sau al principiilor termo-dinamicii etc. Între legile şi principiile existenţei şi legile gândirii umane se interpun activităţile umane, procesele neurofiziologice şi cele psihice, deprinderile de utilizare a semnelor lingvistice, de rostire şi înţelegere, de acceptare şi respingere a diferitelor semnificaţii vehiculate prin intermediul semnelor lingvistice. Geneza formelor şi structurilor logice este, atât la nivelul individului, cât şi la nivelul speciei, intim legată de actele comunicării mediate de semnul lingvistic, cât şi de dinamica actelor comportamentale ca funcţii teleologico-adaptative.

Raportul dintre conţinutul informaţional al unui enunţ şi domeniul realităţii obiective este redat prin valoarea de adevăr a enunţului. Determinarea valorii de adevăr a unei propoziţii date o obţinem cu ajutorul disciplinelor ştiinţifice particulare sau pe baza experienţei practice a subiecţilor cunoscători. Nu este rostul logicii să determine prin sine valoarea de adevăr a propoziţiilor din ştiinţele

[9] V.I. Lenin, *Caiete filozofice*, Editura Politică, p. 64.

[10] Mr. R. Cohen şi E. Nagel, *An Introduction to Logic and Scientific Method*, Harcourt, Bruce, 1934.

experimentale sau întemeiate pe observaţie. Ea ne descrie doar legile sau regulile după care se conservă în consecinţe veridicitatea axiomelor sau premiselor de la care am plecat.

Între logică şi lumea reală se interpun ştiinţele particulare, limbajele acestora şi teoriile dezvoltate în cadrul acestor limbaje. Toate teoriile ştiinţelor particulare folosesc, mai mult sau mai puţin intenţionat, principiile şi legile logicii.

Dintre părţile logicii cea care se ocupă cu raporturile dintre propoziţii şi lumea reală, sau dintre propoziţii şi faptele umane este semantica, ca teorie a semnificaţiilor sau înţelesurilor termenilor şi propoziţiilor. Teoria modelelor logice şi aplicarea logicii într-un domeniu sau altul este in mod major interesată de înmlădierea limbajului logicii la realitatea fizică, economică sau socială. Faptul că logicienii sunt sensibili la aceste imperative ni-l dovedeşte apariţia şi dezvoltarea unor teorii logice intim legate de un aspect sau altul al realităţilor fizice sau sociale. Astfel logicienii se preocupă de fundamentarea teoriilor fizice, de teoria activităţilor umane, de teoria deciziilor, de teoria scopurilor şi programelor, de teoria sistemelor de norme şi a activităţilor normate, de demonstraţia automatizată, de teoria argumentării şi convingerilor etc. Alături de studiul abstract al mecanismelor inferenţiale, se dezvoltă cercetările de logică aplicată, când logicianul întreprinde studiul, cu mijloacele disciplinei lui, al unui segment din realitatea fizică sau socială cu intenţia de a întemeia o strategie eficientă de acţiune.

În viziunea noastră, logica nu este doar o disciplină contemplativă despre propoziţii sau judecăţi, ci un instrument eficient de analiză, formalizare şi modelare a unor variate clase de probleme din toate domeniile cunoaşterii şi din toate domeniile activităţilor practice umane.

6. Adevărul, logica şi teoria cunoaşterii

Noţiunile de adevăr şi de fals apar, deopotrivă, în ştiinţele particulare, matematică, fizică, chimie, drept, dar şi în teoria cunoaşterii şi în logică.

Disciplinele particulare cercetează cu tehnici şi metode adecvate un domeniu din realitatea fizică, biologică sau social istorică, încercând să pună în lumină legi obiective şi raporturi de determinare cauzală. Teoria cunoaşterii descrie, din perspectiva subiectului

cunoscător individual şi colectiv, dinamica procesului cunoaşterii, rolul pe care îl joacă interacţiunea directă dintre subiect şi obiect, observaţia şi experimentul, dar şi rolul semnelor şi limbajului în cunoaştere, natura ipotezelor şi teoriilor ştiinţifice. Categoria centrală a teoriei cunoaşterii este cea de adevăr. Pe filosof îl interesează cum definim adevărul, pe ce căi este acesta accesibil subiectului cunoscător, care sunt resursele şi limitele cunoaşterii, cum putem distinge între adevăr şi fals, între un rezultat corect şi o eroare.

Deşi utilizează la tot pasul conceptele de adevăr şi fals, acestea sunt în teoriile logice concepte primitive. În logica bivalentă admitem că orice propoziţie declarativă are o valoare de adevăr, că este adevărată sau falsă. Propoziţiilor elementare le atribuim, oarecum din oficiu, valori de adevăr. Interesul nostru se concentrează în logică asupra tehnicilor de conservare a adevărului în trecerea de la premise la concluzie, de la axiome la consecinţe. Logica se ocupă cu prioritate de dependenţa valorii de adevăr a enunţurilor complexe, de valoarea de adevăr a enunţurilor elementare şi de dependenţa valorii de adevăr a enunţurilor derivate de valoarea de adevăr a enunţurilor anterior date. Logica nu cercetează în exclusivitate întreaga problematică a adevărului. Definiţia dată logicii de E. Goblot, după care logica cercetează condiţiile cunoaşterii adevărului[11], extinde obiectul logicii în dauna gnoseologiei, fără a aduce prin aceasta nici un folos logicii. Şi teoria cunoaşterii se interesează de organizarea enunţurilor adevărate, de structura lor în ipoteze şi teorii coerente în funcţii descriptive, explicative şi predicative. Gnoseologul poate fi interesat în utilizarea unor tehnici de metode logice, cum ar fi, de exemplu, teoria sistemelor axiomatice, teoria demonstraţiei, teoria interpretării sistemelor axiomatice formale. Dar ţelurile vor fi cu prioritate descrierea procesului cunoaşterii, a căilor şi mijloacelor de atestare şi verificare a autenticităţii cunoştinţelor şi nu teoria inferenţei şi a validităţii.

Dacă este legitim să evidenţiem unghiurile proprii ale gnoseologiei şi ale logicii ar fi, credem, imprudent să aşezăm pietre de hotar severe între graniţele logicii şi ale gnoseologiei, mai ales în condiţiile apariţiei logicilor epistemice şi erotetice, precum şi a cercetărilor logice de inteligenţă artificială, care rezolvă cu mijloace logic-formale, computerizate, unele probleme de teoria demonstraţiei,

[11] E. Goblot, *Traité de logique*, 6-ème ed. A. Clon, Paris, 1937.

aflarea răspunsurilor la unele întrebări, analiza şi sinteza unor programe de calculator, raporturile dintre două grafuri etc.

7. Logica şi teoria limbajelor

Încă de la apariţia sa în cultura europeană, dar şi în cea indiană sau chineză, logica a fost intim legată de analiza expresiilor propoziţionale şi de categoriile gramaticale. Termenii de subiect şi de predicat prezenţi în silogistica aristotelică sunt, în acelaşi timp, termeni de gramatică. La fel, termenul de predicat din logica predicatelor are tot o origine gramatical-lingvistică.

Orice teorie logico-matematică, începând cu logica propoziţiilor şi continuând cu logica predicatelor de ordinul întâi (cu logicile modale, deontice, temporale, dinamice etc.), poate fi prezentată din perspectiva teoriei limbajelor formale şi a gramaticilor generative ca o mulţime de cuvinte într-un alfabet dat, generate de anumite reguli de producţie. În mod analog, demonstraţiile pot fi prezentate ca operaţii de construire a unor texte demonstrative încheiate prin obţinerea tezei de demonstrat.

Aşa, de exemplu, fiind dat un alfabet p, q, r, s... desemnând propoziţii elementare, logica propoziţiilor poate fi descrisă ca o teorie despre conectivele „şi", „sau", „dacă"..., atunci...", „dacă şi numai dacă" etc. care apar în limbile naturale şi în limbajele specializate ale ştiinţelor şi permit formarea propoziţiilor compuse şi calcularea valorii lor de adevăr în funcţie de valoarea de adevăr a propoziţiilor componente. În mod analog, logica predicatelor poate fi privită ca o îmbogăţire a limbajului logicii propoziţiilor prin admiterea în vocabular a constantelor şi variabilelor individuale, a simbolurilor funcţionale şi a simbolurilor predicative, precum şi a cuantificatorilor. Pe această cale, vor putea fi definite noţiunile de termen, de formulă atomară şi cea generică de formulă, precum şi noţiunile de funcţie predicativă şi de propoziţie existenţială sau universală.

În logica predicatelor sunt cercetate, între altele, proprietăţile expresiilor de forma „x este P", „y este în relaţia R cu z", „Există un astfel de x, încât x se află în relaţia S cu z", La fel logica predicatelor ar putea fi caracterizată ca studiul sistematic al cuantificatorilor care apar în limba naturală şi în alte limbaje. Exemple de astfel de cuantificatori sunt „toţi", „unii", „exact", „întocmai", „nici mai mult nici mai puţin", „cel puţin", „mai mult", „cel mult", „mai puţin", „aproximativ", „majoritate", „ minoritate", „puţini", „mulţi". Logica

matematică clasică nu a cercetat decât cuantificatorii existenţial şi pe cei universal redaţi în limba naturală prin „unii" şi „toţi" O analiză mai atentă ne va permite să identificăm mai multe tipuri de cuantificatori, cu domenii de aplicaţie diferite, cu referinţă spaţială sau temporală, obiectivă sau subiectivă.

Fixarea corectă a terminologiei va avea avantajul unei racordări corecte a limbajului logicii predicatelor la limbile naturale şi la limbajele ştiinţelor particulare, la limbajul teoriei agenţilor din inteligenţa artificială.

Există un front larg de lucru în teoria cuantificării, în deosebi în teoria cuantificatorilor inexacţi. Este oportună identificarea mai multor specii de cuantificatori după domeniul lor de aplicaţie. Astfel, cred potrivit să distingem între cuantificatorii temporali ca: „acum", „de acum încolo", „până la data", „începând de la data", „în intervalul ...", „pe durata", cuantificatori ce au fost captaţi în unele sisteme de logică temporală metrizată, cum sunt sistemele create de către N. Rescher, de Johan van Benthem. Cuantificatorii temporali sunt instrumente de referire la date şi durate. Ei sunt instanţiatori sau actualizatori de momente temporale din trecut, prezent sau viitor şi intervin în determinarea actelor de vorbire. La fel cred de actualitate considerarea cuantificatorilor spaţiali ca: „aici", „acolo", „dincolo", „între coordonatele...", „sus", „jos". La fel s-ar putea considera oportunitatea cercetării unor cuantificatori axiologici ca: „foarte slab", „slab", „nici slab, nici bun", „bun", „foarte bun", „mai bun", „mai slab", „excelent", „catastrofal" şi legarea lor de sistemele modale axiologice etc.

Într-un anumit sens, orice teorie logică poate fi privită ca un limbaj artificial sau ca un sublimbaj din limba naturală supus unor exigenţe speciale de construcţie şi controlabilitate. Mai întâi, alfabetul limbajelor logice este mai sărac decât cel al limbilor naturale. În al doilea rând, clasele de expresii sau „categoriile gramaticale" sunt mai puţin numeroase în limbajele logice, reţin sau dau seama de anumite operaţii sau conective, constante sau operatori existenţi în formă „neprelucrată" în alcătuirea limbilor naturale. Orice sistem logic apare astfel ca un limbaj purificat, explicit şi neechivoc, care face transparente alcătuirea expresiilor şi relaţiile dintre formule, permiţând, totodată, prelucrarea formală a acestora. Aşa, de exemplu, relaţia de inferenţă din sistemele logice poate fi pusă în legătură cu procesul de derivare din gramaticile generative, cu derivarea directă şi închiderea tranzitivă.

Teoriile logice au captat treptat diferite conective şi operaţii logice din limbile naturale. Iniţial, teoria logică a cuprins doar aşa-numitele judecăţi asertorice de predicaţie, având structura *S este P*, de exemplu, „Platon este om", „Grecii sunt europeni" Ulterior, prin logica stoică şi megarică au fost identificate şi cercetate într-o primă formă conectivele logic propoziţionale „şi", „sau", „dacă..., atunci...". Judecăţile relaţionale mai complexe nu puteau fi descrise în cadrul logicii clasice aristotelice. Numai elaborarea calculului cu predicate, datorată lui Gottlob Frege şi Ch. Sanders Peirce, în a doua jumătate a secolului al XIX-lea, a făcut posibilă o teorie coerentă despre relaţii şi cuantificatori.

Operatorii modali, temporali şi deontici au fost captaţi în teoriile logice abia în ultimele patru decenii. Putem spune, fără prea mari riscuri, că limba naturală n-a fost complet secătuită de „minereu logic", că asistăm şi în zilele noastre la apariţia unor noi tipuri de logici modale (epistemice, deontice, teleologice), polivalente, preferenţiale, dinamice, temporale etc.

Relaţiile logicii cu lingvistica sunt, în ultima vreme, mereu mai intense. Structuri şi metode logice sunt însuşite şi utilizate de lingvişti. Teoria înţelesului, teoria definiţiilor lexicale şi stipulative, teoria modelelor, teoria ambiguităţii sintactice şi semantice, teoria logică a întrebărilor sau logica erotetică, teoria argumentării, a disputelor şi negocierilor sunt, deopotrivă, subiecte de interes pentru logicieni şi pentru lingvişti. Conceptele de limbaj şi de gramatică pot fi descrise în cadrul unor programe logice. Pe de altă parte, multe capitole noi ale logicii au apărut din studierea cu mijloace specifice a unor conjuncţii sau prepoziţii din limbile naturale, cărora li s-au asociat contexte pragmatice şi interpretări semantice adecvate.

8. Gândirea logică, timpul şi activităţile umane

Am raportat până acum operaţiile logice la evenimentele fizic-naturale, la actele de rostire şi comunicare, la operaţiile şi procesele de cunoaştere. Este firesc să ne întrebăm dacă nu cumva operaţiile logice au vreo legătură cu conceperea şi realizarea activităţilor practice umane. Este mai presus de orice îndoială că, atât operaţiile şi activităţile practice umane, actele de rostire, cât şi demersurile logice sunt secvenţe sau şiruri de transformări organizate diacronic şi sincronic. Putem concepe două operaţii practice executate simultan, dar şi două operaţii practice executate succesiv. La fel, putem concepe două propoziţii

acceptate simultan, după cum putem concepe ordonarea lor diacronică, cum se întâmplă de fapt cu propoziţia concluzie şi premisele din care a fost derivată aceasta. În actul real de gândire logică, concluzia este întotdeauna ulterioară premiselor, deşi în logica formală, clasică, facem întotdeauna abstracţie de timp. Acceptăm, oarecum tacit, supoziţia că, de îndată ce am acceptat premisele unui raţionament, dobândim instantaneu şi consecinţele cele mai îndepărtate ale acestora, ca şi cum înaintarea noastră spre consecinţe s-ar face cu viteza luminii sau instantaneu. În logica formală clasică durata travaliului intelectual este contractată până la anihilare. De îndată ce am acceptat axiomele sau premisele şi cunoaştem regulile de inferenţă, am acceptat şi teoremele sau consecinţele. Din nefericire, operaţiile reale efectuate de om sau de un computer nu sunt atemporale, instantanee şi chiar în ipoteza că există un algoritm pentru demonstrarea unei anumite teoreme sau pentru aflarea răspunsului la o întrebare, descoperirea şi executarea acestuia se face în timp, prin încercări şi erori pentru problemele nerezolvate şi prin aplicarea unor reguli sau executarea unui program pentru problemele de rutină.

Cercetările de psihologie genetică, ca şi cele de praxiologie, au pus în lumină, apropierea dintre actele de gândire logică şi operaţiile sau activităţile umane practice. Fără a ne propune aici cercetarea genezei operaţiilor şi structurilor logice la subiectul uman, credem că apariţia acestora este o consecinţă a dezvoltării aptitudinilor acţional-performative şi lexical-discursive. Dobândirea aptitudinilor fiinţei umane de construcţie şi discurs logic, este, într-un anumit sens, o consecinţă a achiziţiei deprinderilor de înlănţuire coerentă a operaţiilor şi activităţilor practice, de adecvare a mijloacelor la scopuri sau o conduită teleologică, raţională şi o consecinţă a deprinderii de codificare natural-lingvistică a experienţelor indivizilor umani. Înainte de a descrie raporturi atemporale între valorile de adevăr ale enunţurilor, construcţia „dacă..., atunci..." a descris raporturi de succesiune între operaţiile sau acţiunile umane şi rezultatele acestora. La fel înclinăm să credem că particulele „şi", „sau", „dacă şi numai dacă", au descris iniţial acte sau operaţii practice şi numai ulterior raporturi între valorile de adevăr ale enunţurilor ce le descriu. Căutarea unei demonstraţii este o conduită teleologică sau o secvenţă de acte intenţionale, care presupun întotdeauna o fază reductiv-analitică şi o fază sintetic-constructivă.

Trebuie să admitem că există o legătură organică între succesiunea actelor de conduită umană, inclusiv între scop, conduită şi rezultat, pe de o parte, şi succesiunea actelor de vorbire comunicare şi înţelegere în procesele de cooperare interumană, pe de altă parte. Apariţia gândirii logic-discursive nu poate fi ruptă de activităţile practice umane şi de logica raporturilor dintre scopuri şi mijloace.

9. Descriptiv şi normativ în logică

Pentru unii autori, logica nu este o disciplină ştiinţifică, ci o artă, o metodă sau o disciplină normativă. Aşa, de exemplu, în logica de la Port Royal (1662), întocmită de Antoine Arnold şi Pierre Nicole, logica este definită ca *arta conducerii corecte a raţiunii unei persoane în cunoaşterea lucrurilor*, deopotrivă în instrucţia proprie şi în instrucţia altora. Cu ajutorul logicii, noi trebuie să dobândim *claritatea expunerii*. Logica se ocupă de modul *cum concepem noţiunile*, de *alcătuirea judecăţilor* din noţiuni şi a raţionamentelor din judecăţi, dar şi de *ordonarea raţionamentelor* noastre după o anumită strategie. Logica cuprinde, aşadar, şi metoda de analiză şi expunere a problemelor. Logica de la Port Royal sprijină într-un anumit sens metoda deductivă carteziană şi are reţineri faţă de inducţie şi metodele empirico-experimentale promovate de ştiinţele naturii.

Logica este o *disciplină metodologică preliminară*, o *propedeutică*, ce trebuie să preceadă studiul altor discipline. Ea este *un instrument intelectual universal*, care educă raţiunea subiecţilor cunoscători, le formează deprinderea de *a concepe şi defini clar*, de a judeca şi raţiona corect, de a cerceta cu ordine şi eficienţă. Logica nu e numai o ştiinţă, dar şi *o abilitate, o măiestrie, o artă*. Viziunea şcolii de la Port Royal a persistat în vreme. Pentru John Stuart Mill, unul dintre teoreticienii metodelor inductive, logica este o ştiinţă a raţionamentului şi o artă întemeiată pe această ştiinţă.[12] Ideea logicii ca o artă este prezentă şi în scrierile logicianului neotomist Jacques Maritain: „Logica este arta care ne dezvoltă capacitatea de a proceda în ordine, accesibil şi corect în activitatea raţiunii însăşi"[13]

[12] John Stuart Mill, *A System of Logic*, London, 1843, p. 227.

[13] Jaques Maritain, *An Introduction to Logic*, London Sheed and Ward, 1937, p. XIII.

Autorii care privesc logica ca o artă o înţeleg ca o aptitudine sau abilitate a intelectului sau raţiunii umane. Această abilitate nu este înnăscută, ci dobândită prin instrucţie de subiectul cunoscător. Distincţia dintre legile gândirii corecte exprimate în formalismele logice şi însuşirea de către subiectul uman concret a legilor gândirii corecte, a abilităţii şi deprinderii de a conchide valid din premise adevărate concluzii adevărate este cât se poate de legitimă. Asimilarea deprinderilor de a defini, clasifica şi deduce corect este rezultatul unui efort de învăţare a logicii. Subiectul cunoscător trebuie să deţină tehnica şi arta de a gândi corect. Dar, orice disciplină ştiinţifică, nu numai logica, putem să o surprindem la nivelul structurilor sale impersonale, obiectivate în demonstraţiile din tratate şi la nivelul asimilării ei de către subiectul cunoscător individual, utilizatorul sau beneficiarul ştiinţei.

Dintre toate disciplinele ştiinţifice, logica îşi evidenţiază, poate cel mai pregnant, dimensiunea sa *metodologică, de îndreptar sau călăuză în cunoaştere.* Ea este după W. Wundt, *ştiinţa legilor normative ale cunoaşterii omeneşti*[14]. Ideea provine din concepţia lui Kant, după care „logica este o ştiinţă a raţiunii, referitoare nu la materie ci la forma pură[15]; o ştiinţă a priori despre legile necesare ale gândirii, dar nu cu privire la obiecte particulare, ci la toate obiectele în genere; deci o ştiinţă a aplicării corecte a intelectului şi raţiunii în genere, dar nu în mod subiectiv, adică nu pe baza principiilor empirice (psihologie) cum gândeşte intelectul, ci în mod obiectiv, adică pe baza principiilor a priori, aşa cum ar trebui să gândească intelectul". *Logica este un canon al intelectului şi al raţiunii,* dar numai cu privire la ceea ce-i formal în folosirea lor, ori care ar fi conţinutul. În logică se face abstracţie de orice conţinut al cunoaşterii intelectului şi al diversităţii obiectelor, reţinându-se doar forma gândirii. Logica pură nu scoate nimic din psihologie, care, deci, nu are nici o influenţă asupra canonului intelectului [15]

Deci, după Immanuel Kant, *logica este o ştiinţă despre formele pure ale intelectului şi ale raţiunii,* forme la care accedem prin abstracţie, prin abandonarea conţinutului concret şi a experienţei

[14] W. Wund, *Logik,* p. 7.

[15] Immanuel Kant, *Critica raţiunii pure,* Editura Ştiinţifică, Bucureşti, 1969, p. 93; vezi şi Immanuel Kant, *Logica generală,* Editura Ştiinţifică şi Enciclopedică, Bucureşti, 1985, p. 69.

[15] Idem.

empirice şi prin depăşirea particularităţilor psihologice, subiective ale dobândirii cunoştinţelor.

După Kant, logica este o disciplină asertoric-descriptivă, care cercetează legile necesare ale gândirii în structura lor formală, dar şi o disciplină normativă, care dezvăluie canoanele intelectului, regulile cărora li se supune acesta.

10. Niveluri ale demersului logic şi accepţii ale termenului logică

Obiectul cursului nostru este ştiinţa logicii. Admitem cu toţii că geologia se ocupă de formarea şi structura rocilor din care este alcătuit pământul; botanica se ocupă de plante, zoologia studiază animalele. Fiecare dintre aceste discipline studiază un domeniu din realitatea fizică înconjurătoare. Există oare vreun domeniu din lumea înconjurătoare de care se ocupă logica sau logica, spre deosebire de disciplinele citate mai sus, nu cercetează o realitate fizică independentă de om şi de activitatea sa.

Trebuie să admitem că logica nu se ocupă de nivelul ontic primar, ca fizica, mineralogia, botanica sau zoologia. Ea nu se ocupă nici de existenţa artificială, creată de oameni pe parcursul existenţei lor sociale, în activităţile lor productive, cum sunt ştiinţele tehnice, de unelte, materii prime, instalaţii, maşini, surse de energie, etc. Logica se ocupă de anumite particularităţi ale gândirii umane în procesul utilizării limbilor naturale pentru a face declaraţii sau aserţiuni despre lumea înconjurătoare şi a descoperi adevărurile transmise prin aceste declaraţii. Mai mult, ea se ocupă de modul cum putem noi prelucra o mulţime de propoziţii declarative astfel încât să nu trecem niciodată de la propoziţii adevărate la propoziţii false.

Logica, precum şi matematica, presupune un proces inferenţial, o prelucrare sau un calcul, prin care se conservă anumite adevăruri de la o stare anterioară la o stare ulterioară. Demersul logic se situează la nivele diferite ale discursului raţional utilizat de fiinţa umană.

Pentru a distinge mai uşor nivelurile şi accepţiile termenului logică în viaţa cotidiană să considerăm pentru început câteva propoziţii:

1. Cuvântarea Dv. despre dreptul de proprietate al cetăţenilor este logică şi convingătoare.
2. Articolul lui Petru are o logică imbatabilă.
3. Planul Dv. este realist şi logic.

4. Logica aristotelică este expusă în „Analiticele prime" şi în „Analiticele secunde"

5. Logica predicatelor a fost creată autonom de către Gottlob Frege şi de Ch. S. Peirce.

6. Sistemul axiomatic Hilbert–Ackermann este necontradictoriu.

7. Logica predicatelor de ordinul întâi este indecidabilă.

Prima propoziţie se referă la înlănţuirea coerentă şi convingătoare a ideilor în discursul unui vorbitor. Proprietăţile sau calităţile logice, coerenţa, forţa de convingere sunt atribuite *discursului oral* al unui orator care a vorbit în faţa noastră despre dreptul la proprietate al cetăţenilor unui stat, de exemplu, al cetăţenilor unui fost stat comunist.

Cea de a doua propoziţie se referă la logica şi coerenţa unui *discurs scris,* a unui articol, să spunem, publicat într-un ziar de un autor oarecare, pe nume Petru.

Cea de a treia propoziţie atribuie calităţi logice nu doar unor rostiri sau scrieri oarecare, ci rostirilor sau scrierilor ce descriu planuri şi, deci, prin consecinţă, planurilor ca ordonări de conduite umane prin care se ating anumite stări posibile anterior asumate de cineva ca scop.

Cea de a patra propoziţie nu se referă la demersul logic al unei fiinţe umane despre un subiect oarecare ci la o *teorie logică particulară*, creată de un filosof, Aristotel, la teoria judecăţilor de predicaţie de forma S este P şi la raţionamentele valide ce pot fi făcute cu acest fel de judecăţi supuse unor restricţii, la silogismele categorice, ca şi la silogismele modale. Propoziţia ne spune în ce scrieri a fost expusă teoria logică aristotelică. O teorie logică poate fie redată prin anumite principii, legi şi reguli ce descriu prelucrări corecte, valide ale formulelor iniţiale ce descriu declaraţii sau aserţiuni.

Cea de a cincea propoziţie vorbeşte despre o altă teorie logică, una de bază pentru ştiinţa actuală a logicii, despre logica predicatelor şi ne informează cine au fost creatorii ei.

Cea de a şasea propoziţie vorbeşte despre un sistem logic axiomatic de logica propoziţiilor, cel creat de David Hilbert şi W. Ackermann, despre care ne spune că este necontradictoriu.

Cea de a şaptea propoziţie ne vorbeşte de o proprietate a unei teorii logice şi anume despre proprietatea logicii predicatelor de a fi indecidabilă.

Propoziţiile 6 şi 7 se referă la teorii logice ca amsamblu de enunţuri despre legile logicii despre care se enunţă anumite proprietăţi. Acestea sunt *propoziţii despre sisteme de propoziţii* şi vizează aşa numitul *nivel metateoretic* al unui discurs.

Trebuie, aşadar, să distingem între: 1. logica unui discurs uman sau logica unui emitent sau orator; 2. structura logică a unei scrieri a unui autor; 3. logica unui plan de acţiune care vizează o anumită ordine firească a înlănţuirii unor operaţii sau conduite în realizarea unui scop asumat de un agent individual sau colectiv; 4. expunerea concepţiei unui autor despre o teorie logică sau expunerea unei teorii logice, cum ar fi logica aristotelică, logica stoică, logica propoziţiilor, logica predicatelor, logica deontică etc. (vezi propoziţiile 4 şi 5); 5. reflecţia metateoretică despre teoriile logice (vezi propoziţiile 6 şi 7), gândirea despre teoriile logice, metalogica înţeleasă ca o teorie despre sistemele logice şi proprietăţile lor formale, cum ar fi consistenţa sau noncontradicţia, completitudinea, independenţa, decidabilitatea.

Exemple de teorii logice sunt: logica aristotelică, logica stoică, logica propoziţiilor, logica predicatelor de ordinul întâi sau de ordinul al doilea, logica modală aletică, logica deontică, logica epistemică, logica temporală, logica polivalentă, logica nemonotonă etc.

O *teorie* logică are o sintaxă sau un limbaj formal şi o semantică. Pentru o teorie logică sunt esenţiale noţiunile de *formulă bine formată, propoziţie, formulă realizabilă, formulă infirmabilă, tautologie sau lege logică şi contradicţie logică sau formulă irealizabilă, regulă de deducţie, deducţie şi deducţie validă.*

Logica poate fi înţeleasă şi ca o *metodologie*, ca un ansamblu de reguli şi tehnici în conformitate cu care se organizează un discurs uman sau o activitate umană, indiferent de domeniul de referinţă al acestora. Putem, aşadar, vorbi de logică ca *metodă şi tehnică de control a corectitudinii construcţiilor teoretice* şi ca *instrument de rezolvare a problemelor* cu care ne confruntăm în viaţa reală. În zilele noastre logica tinde să devină o tehnică raţională de analiză, formalizare şi modelare a rezolvării computerizate a problemelor. Logica asistă, în calitate de arhitect, la alcătuirea unui plan complex multimodular de înţelegere şi rezolvare practică a problemelor, căci ea intervine din ce în ce mai mult în alcătuirea bazelor de cunoştinţe şi în definirea maşinii inferenţiale prin care acestea pot fi rezolvate. Logica devine un instrument eficace.

Logica este tot mai des privită de informaticieni ca un producător de tehnici şi metode intelectuale. Între principalele metode

utilizate în logică sunt formalizarea, axiomatizarea, modelarea, metode de decizie în diferite sisteme şi teorii logice, cum sunt, metoda matriceală, metoda tablourilor semantice, metoda arborilor de decizie, metoda lui Quine, metoda Davis-Putnam, metoda rezoluţiei, diferite strategii rezolutive. La acestea trebuie adăugate tehnici şi metode utilizate în logica acţiunii, în teoria inducţiei, în raţionamentele abductive, în cele plauzibile sau prin analogie etc.

Putem desigur să ne întrebăm dacă există o structură logică a discursului interior tacit al unui subiect care receptează un mesaj, care analizează şi decodifică un mesaj, face un calcul sau un raţionament tacit, neexprimat verbal, îl evaluează critic, îl acceptă sau îl respinge şi îşi ticluieşte în sinea sa o replică sau un răspuns la întrebarea sau aserţiunea unui interlocutor al său. Noi credem că subiectul cunoscător săvârşeşte operaţii şi calcule logice şi la nivelul limbajului interior propriu, la nivelul ideolectului, că acestea au o dinamică proprie şi că ele pregătesc discursul verbal explicit al agentului. Mai mult, credem că pot fi concepute experimente mai subtile de psihologia cunoaşterii sau de ştiinţele cognitive întemeiate pe logicile epistemice şi doxastice.

Observaţiile de mai sus ne permit să conchidem că: 1. principiile şi schemele gândirii logice intervin la nivelul fiecărei fiinţe umane, mai mult sau mai puţin conştient; 2. Nici măcar gândirea umană tacită nu este complet străină de legile şi principiile logicii preluate din relaţiile de ordine impuse de actele şi conduitele umane sau din structurile lingvistice încorporate în ideolectul sau limbajul individual al subiectului cunoscător sau al oratorului; 3. Structuri logice sunt încorporate în toate limbile naturale, în toate limbajele artificiale şi în toate sistemele de norme sau reguli formulate pentru un compartiment sau altul al activităţilor umane; 4. Orice teorie ştiinţifică care identifică principii şi care degajează enunţuri adevărate sau teoreme din aceste principii conţine inevitabil un apel la legi şi scheme logice de inferenţă; 5. Orice discurs public apelează la un fond prealabil de cunoştinţe admis de interlocutori, emite şi susţine aserţiuni sau teze şi respinge sau combate alte aserţiuni, încercând să întemeieze deopotrivă afirmaţiile precum şi respingerile; 6. Planurile şi programele de acţiune îmbracă tot mai des forma unor argumentări procedurale şi sunt adesea susţinute cu tehnici şi procedee computaţionale.

11. „Harta logicii" şi direcţii ale cercetării logice contemporane

În 1974, profesorul ieşean Petre Botezatu publica un studiu „Harta logicii"[16], în care, pornind de la două studii ale lui N. Rescher[17], marca conturul provinciilor mai vechi şi mai noi ale împărăţiei logicii. Erau enumerate întâi mai multe tipuri de cercetări logice privind *logica de bază* (logica tradiţională aristotelică, logica matematică de orientare clasică, îndeosebi, logica propoziţiilor şi logica predicatelor, logica matematică neortodoxă sau logicile „deviante", cum ar fi, de exemplu, logicile intuiţioniste, polivalente şi constructiviste, care resping unele teze ale logicii clasice (bivalenţa, terţiul exclus, dubla negaţie etc.) sau logicile temporale şi modale care adaugă operatori şi teze noi pe temelia logicii matematice clasice. Apoi, erau menţionate cercetările de *metalogică* în deosebi, cele de sintaxă, semantică şi pragmatică şi, în sfârşit, cercetările cu valoare aplicativă sau „dezvoltările" logice din matematică, fizică, biologie, ştiinţe sociale, lingvistică şi filosofie.

„Hărţile" întemeiate în 1968 şi 1974, chiar şi de către cartografi cu intuiţie şi perspicacitate, sunt în mare parte depăşite în 1999.

Au apărut insule sau arhipelaguri noi: logicile dinamice create de V.R. Pratt şi dezvoltate de D. Harel, K. Segerberg, P. Pnueli, logica fuzzy creată de Zadeh şi dezvoltată de A. Kandel, de M. Mizumoto, R.C.T. Lee, logicile intensionale şi relevante dezvoltate de A.R. Anderson, N. D. Belnap, J. Hintikka, E. K. Voişvillo, A. Sidorenko.

Cercetarea logică contemporană pare în prezent deopotrivă stimulată de reflecţia filosofică şi am menţiona în această direcţie contribuţiile majore ale unor filosofi ca G. H. von Wright, creatorul logicilor deontice şi al unor versiuni de logica acţiunii, A. N. Prior, creatorul logicilor temporale, J. Hintikka, creatorul logicii epistemice şi doxastice sau S. A. Kripke, unul dintre creatorii semanticii lumilor posibile şi de contactul cu ştiinţele particulare îndeosebi cu ştiinţa computerelor. Din contactul logicii modale cu limbajele de programare şi cu nevoia elaborării unor semantici pentru aceasta s-au născut, pe de o

[16] Petre Botezatu, *Semiotică şi negaţie*, Editura Junimea, 1974, p. 37-51.

[17] Nicholas Rescher, „Recent Developments în Philosopical Logic" şi „Discours and Method" (1959) în *Topics in Philosophical Logic*, Holland, 1968, p. 1-23, 332-341.

parte, cercetările de logică dinamică. Acelaşi contact cu ştiinţa calculatoarelor, a dus, pe de altă parte, la cercetări de teoria raţionamentelor nuanţate (Gr. C. Moisil), la apariţia teoriei mulţimilor fuzzy şi a logicii fuzzy.

De mare interes sunt astăzi cercetările de logică nemonotonă dezvoltate de Drew McDermott, care îşi propun să dea seama de dinamica actelor de cunoaştere şi să poată descrie revizuirea bazelor de cunoştinţe[18].

Extinderea ariei de aplicaţie a calculatoarelor în domeniile industriei şi gestiunii economice, în cercetarea ştiinţifică şi în proiectare, apariţia programelor expert şi a inteligenţei artificiale au ridicat noi probleme cercetării logice, legate în deosebi de teoria funcţiilor recursive, a algoritmilor şi a calculabilităţii şi, în genere, de teoria demonstraţiei. Metoda rezoluţiei creată de J. A. Robinson şi dezvoltată de D. W. Loveland, N. Nilsson şi alţii, a permis trecerea la prelucrarea bazelor de cunoştinţe cu mijloace logico-inferenţiale, demonstrarea automată a unor teoreme, extrageri de răspunsuri la întrebări dintr-o bază de cunoştinţe etc. Două milenii logica a cercetat în exclusivitate doar enunţurile asertorice, declarative, întrebările; ordinele sau propoziţiile prescriptive şi normele ce reglementează juridic, moral sau tehnic diferite laturi ale activităţilor umane fiind eliminate din câmpul cercetării logice ca nerelevante. De mai bine de trei decenii, logica deontică cercetează propoziţiile prescriptive şi normele ce reglementează juridic, moral sau tehnic diferite laturi ale activităţile umane.

Logica erotetică s-a conturat tot de trei decenii, ca o disciplină logică distinctă menită să cerceteze propoziţiile interogative şi conceptul de problemă. La conturarea acestui domeniu şi-au adus contribuţia A. N. Kolmogorov, E. Sperantia, L. Aqvist, J. Hintikka, M. Bunge, P. Botezatu[19] Nevoile descrierii roboţilor industriali, cerinţa ordonării în timp a operaţiilor ce urmează să le execute, a condus la creşterea interesului programatorilor pentru conceptele de acţiune, pentru ideile de scop, mijloc, accesibilitate, program, eficacitate şi eficienţă, toate concepte ce ţin de praxiologie sau de teoria acţiunii, dar pot fi, în acelaşi timp, caracterizate cu mijloacele logicilor modale şi temporale.

---

[18] Drew McDermott and Jon Doyle, *Non-Monotonic Logic I, Artificial Intelligence*, 13 (1980), 41-72.

[19] Vezi *Logica interogativă şi aplicaţiile ei*, Editura Ştiinţifică şi Enciclopedică, 1982, selecţie, traducere şi studiu introductiv de prof. univ. dr. Constantin Grecu.

Tot mai multe teorii logic formale elaborate de specialişti în programare şi robotică, J. Mc. Charty, R. Moore, A. R. Hook, V. Lifschitz, K. Konolige, M. J. Wooldridge, N. R. Jennings, W. Balzer încearcă să descrie interacţiunea dintre acţiune şi cauzalitate, dintre cunoaştere şi acţiune, dinamica acestora.

Sunt numeroase tentative de a elabora teorii logico-matematice, despre mecanismul luării deciziilor, despre condiţii, stări alternative, criterii şi rezultate. Logica preferinţelor dezvoltată de S. Halden, B. Hanson, R. M. Martin, R. M. Chisholm, E. Sosa, von Wright, R. Toumela, sunt doar momente şi realizări parţiale într-o tendinţă mai generală de a contura o teorie multidimensională a conduitelor uman raţionale. Logica este chemată în epoca noastră din ce în ce mai mult să întemeieze nu doar concluziile noastre într-un discurs teoretic, dar şi deciziile noastre într-o activitate practică de extensiune şi implicaţii sociale tot mai vaste.

12. Concluzii

O istorie de peste două milenii a conferit termenului de logică accepţii extrem de diferite:

1. teorie despre deducţie şi inferenţă, despre definiţii şi clasificări, despre operaţiile de prelucrare a informaţiei în limbile naturale şi artificiale în condiţiile conservării în consecinţe a adevărului din propoziţiile iniţiale şi a funcţiilor denotative ale termenilor sau descripţiilor;
2. teorie a raţionamentului silogistic;
3. teorie a funcţiilor de adevăr;
4. teorie a consecinţelor logice semantice şi a modelelor semantice;
5. teorie despre operaţiile şi demersurile gândirii umane coerente;
6. teorie despre legile şi structurile cele mai generale ale naturii, societăţii şi gândirii;
7. artă a conducerii corecte a raţiunii unei persoane în cunoaşterea lucrurilor;
8. teorie a argumentării sau teorie despre întemeierea concluziilor noastre şi despre întemeierea deciziilor şi conduitelor umane;
9. ştiinţă a legilor normative ale cunoaşterii omeneşti;
10. ramură a matematicii chemată să fundamenteze raţionamentele din disciplinele matematice, care se află în acelaşi raport cu analiza şi critica gândirii, precum geometria cu ştiinţa spaţiului;

11. teorie a bazelor de cunoştinţe şi a demonstraţiei automatizate;
12. teorie a raţionalităţii în discurs, decizie, program şi conduită.

Având în cultura europeană o vechime de peste două mii trei sute de ani (Aristotel 384-322 î.Hr.), logica a fost la început o teorie a noţiunilor, judecăţilor şi raţionamentelor de tip silogistic, asertorice şi modale. Logica clasică a cercetat cu prioritate aşa-numitele judecăţi de predicaţie, de forma *S este P*. Odată cu stoicii au fost cercetate conectivele logice, implicaţia, conjuncţia şi disjuncţia, evidenţiindu-se dependenţa valorii de adevăr a propoziţiilor complexe de valoarea de adevăr a propoziţiilor elementare componente.

Utilizarea metodelor algebrice în studiul logicii o datorăm lui George Boole (1815-1864) iar crearea primelor sisteme axiomatice pentru logica propoziţiilor şi logica predicatelor, o datorăm lui Gottlob Frege (1848-1925). Primul tratat de logică matematică este *Principia Mathematica*,1910-1913, opera lui A. N. Whitehead şi B. Russell.

Prin C. I. Lewis, logica matematică s-a extins la teoria modalităţilor, iar prin Post şi J. Lukasiewicz, teoria logică modernă s-a extins la logicile polivalente, cu mai multe valori, ternare, cuaternare.

Extinderea obiectului logicii nu a avut loc doar în plan orizontal prin adăugarea de noi domenii de investigaţie, dar şi în plan vertical, prin trecerea de la nivelul limbajului obiect la metalimbaj, la reflecţia despre structura şi proprietăţile formale ale limbajului obiect.

Logica şi-a îmbogăţit şi perfecţionat permanent tehnicile şi metodele de investigaţie. Cultura logică nu presupune doar asimilarea de teorii şi rezultate logice, mari teoreme, dar şi *asimilarea de tehnici şi metode*. În prezent, ştiinţa logicii utilizează metode formale, algebrice, topologice, metode semantice, metode algoritmice, care pun în lumină proprietăţi ale formulelor logice, ale legilor sau tautologiilor, ale sistemelor axiomatice sau determină raporturile dintre diferitele teorii logice.

Asimilarea tehnicilor şi metodelor logice nu se dovedeşte utilă doar pe teritoriul logicii ca ştiinţă, dar şi pe teritoriul altor discipline şi domenii ale activităţilor umane. *Valoarea logicii rezidă în funcţia ei formativă şi organizatoare, în ordonarea şi relevarea corelaţiilor dintre probleme, în evidenţierea proprietăţilor unor algoritmi de rezolvare*. Logica este, chiar de la crearea ei, un organon sau un instrument de analiză şi prelucrare a discursului teoretic, raţional şi, în epoca noastră, un mijloc de întemeiere raţională a deciziilor şi conduitelor umane.

13. Teme şi probleme de dezbatere. Posibile subiecte de examen

1. Logica aristotelică şi logica matematică.
2. Componentele şi structura unei teorii logice. Sistem şi teorie logică.
3. Principiile logicii în teoria aristotelică şi în logica predicatelor.
4. Clasificarea teoriilor logice din perspectiva principiilor logice.
5. Alfabet şi reguli de formare. Limbajul formal al unei teorii logice.
6. Limbile naturale şi limbajele formale. Formalizarea şi interpretarea unui limbaj formal.
7. Logica şi teoria deducţiei. Schemele de inferenţă şi raţionamentele valide.
8. Relaţia de consecinţă logică. Consecinţă logică sintactică şi consecinţă logică semantică.
9. Logic şi ontologic. Legile logicii şi legile naturii.
10. Logica şi teoria cunoaşterii. Logica şi teoriile ştiinţifice.
11. Raporturile dintre logică şi psihologie. Principiile logicii şi gândirea umană coerentă.
12. Logica modernă şi ştiinţele cognitive
13. Logica şi limba naturală. Conective, cuantificatori şi operaţii logice în limbile naturale
14. Ontologia, teoriile ştiinţifice şi semantica logică.
15. „Harta" disciplinelor logice. Logicile deviante şi logicile nedeviante.
16. Logica şi demonstraţia automatizată.

14. Bibliografie

1. Petre Botezatu, *Constituirea logicităţii*, Editura Ştiinţifică şi Enciclopedică, 1983, p. 95-101.
2. Petre Botezatu, *Semiotică şi negaţie*, Editura Junimea, 1974, p. 37-51.
3. * * * *Logică şi filozofie*, Editura Politică, 1968.
4. Gh. Enescu, *Tratat de logică*, Editura Lider, 1997.
5. Cornel Popa, *Logica predicatelor*, Editura Hyperion, 1992.
6. Cornel Popa, *Logica simbolică şi bazele de cunoştinţe*, I, II, Universitatea „Politehnica" Bucureşti, 1998, p. 19-40.
7. G. Metakides, A Nerode, *Principii de logică şi programare logică*, Editura Tehnică, 1998.

SECȚIUNEA I

LOGICA PROPOZIȚIILOR

Cap. 1. SINTAXA ŞI SEMANTICA LOGICII PROPOZIŢIILOR

1. Propoziţie şi semn propoziţional

Prin *propoziţie* sau judecată în logica propoziţiilor se înţelege o formulă bine formată în limbajul formal al logicii propoziţiilor. Ea exprimă de regulă o declaraţie sau un enunţ rostit în viu grai sau scris într-o limbă naturală (sau în limbajul unei ştiinţe), o expresie cu înţeles ce poate fi adevărată sau falsă. Propoziţiile descriu în mod curent evenimente sau fapte din lumea reală, dar pot descrie şi stări sau evenimente imaginate de fiinţele umane.

Logica clasică vizează aşa-numitele propoziţii *asertorice* sau *declarative.* Ea omite din sfera sa de cercetare propoziţiile interogative, prescriptive, propunerile, rugăminţile sau promisiunile care sunt cercetate de alte teorii logice. În logica propoziţiilor se face abstracţie şi de structura internă a propoziţiilor. Se cercetează în exclusivitate modul de agregare a propoziţiilor, obţinerea propoziţiilor compuse din propoziţii elementare. Problema centrală a logicii propoziţiilor este evidenţierea dependenţei valorilor de adevăr ale propoziţiilor compuse, de valorile de adevăr ale propoziţiilor elementare.

În logică facem distincţie între *semnul* sau *expresia propoziţională* şi *conţinutul informaţional* fixat, păstrat şi transmis prin semnul propoziţional. Frecvent, într-o limbă acelaşi conţinut informaţional poate fi transmis prin semne diferite. Aşa de exemplu, enunţurile:

1. Aristotel a fost profesorul lui Alexandru Macedon
2. Filosoful din Stagira a fost profesorul lui Alexandru Macedon.
3. Autorul *Analiticelor prime* a fost profesorul urmaşului lui Filip la tronul Macedoniei.

Cele trei propoziţii transmit, în esenţă, aceeaşi judecată, după care regele Macedoniei, Alexandru cel Mare, l-a avut ca profesor pe filosoful Aristotel, autorul *Analiticelor prime.*

Expresiile nominale „Aristotel", „filosoful din Stagira" şi „autorul *Analiticelor prime*" vizează aceiaşi fiinţă umană care a trăit în Grecia Antică între 383–322 î.e.n. şi între altele a creat silogistica.

La fel „Alexandru Macedon" şi „urmaşul lui Filip la tronul Macedoniei" vizează acelaşi personaj istoric care a trăit în Grecia între 356–323 î.e.n.

La un alt nivel de analiză, în teoria dialogului sau în logica epistemică şi pragmatică cele trei enunţuri de mai sus nu vor mai fi echivalente. Diferiţi subiecţi cunoscători le vor putea înţelege şi interpreta în mod diferit.

Din perspectiva primului capitol al logicii matematice – logica propoziţiilor – cele trei enunţuri de mai sus desemnează aceiaşi propoziţie logică sau aceiaşi judecată şi sunt, împreună, adevărate sau false. Mai mult, chiar dacă vor fi notate cu variabile propoziţionale diferite, când vor fi interpretate li se va atribui aceeaşi valoare de adevăr.

2. Limbajul logicii propoziţiilor

După cum am văzut orice teorie logică riguroasă presupune un limbaj formal propriu şi un ansamblu de reguli semantice cu ajutorul cărora pot fi definite noţiunile de validitate, realizabilitate şi consecinţă logică.

Pentru definirea limbajului formal al logicii propoziţiilor avem nevoie de un alfabet şi de un număr de reguli de bine formate a formulelor admise în limbaj.

Alfabetul este un set de semne distincte menite să desemneze clase de obiecte primitive presupuse sau admise în teoria în cauză. De regulă, mulţimea semnelor ce intervin într–un alfabet este scindată în mai multe subalfabete sau clase de simboluri primitive. Astfel în alfabetul logicii propoziţiilor vor interveni trei clase distincte de simboluri primitive: 1. *variabile propoziţionale* sau simboluri pentru propoziţiile elementare; 2. *conective logice* sau simboluri ce mijlocesc agregarea formulelor elementare sau compuse în formule mai complexe ce le conţin pe primele; 3. *semne de grupare* sau paranteze:

$A1 = [\ p, q, r, s, t, v, w]$	*variabile propoziţionale*
$A2 = [\ -, \wedge, \vee, \supset, \equiv, /\ ]$	*conective logice*
$A3 = [\ (\ ,\ )\ ]$	*semne de grupare*
$A = A1 \cup A2 \cup A3$	*alfabetul logicii propoziţiilor*

Definim mulţimea *şirurilor finite de simboluri de lungime k*, scrise în alfabetul A, prin:

$$A^* = \cup a^k : a \in A,\ 1 \leq k < \infty$$

Este important de subliniat că nu orice şir finit din A* va fi o formulă bine formată în limbajul logicii propoziţiilor. Vor fi formule bine formate doar acele şiruri finite de simboluri care pot fi formate cu elemente din A1 cu ajutorul conectivelor logice (unare sau binare) enumerate în mulţimea A2 folosind anumite *reguli de bine formare.* Dăm mai jos aceste reguli:

P1 Dacă $\alpha \in$ A1, mulţimea variabilelor propoziţionale, atunci α este o formulă bine formată în limbajul logicii propoziţiilor.

P2 Dacă α este o formulă bine formată, atunci $-\alpha$ va fi, deasemenea, o formulă bine formată.

P3 Dacă α şi β sunt formule bine formate în limbajul logicii propoziţiilor, atunci:

$\alpha \wedge \beta$, $\alpha \vee \beta$, $\alpha \supset \beta$, $\alpha \equiv \beta$, vor fi şi ele formule bine formate în limbajul logicii propoziţiilor.

Putem, acum, defini formal limbajul logicii propoziţiilor. Numim *limbaj al logicii propoziţiilor* mulţimea şirurilor de simboluri scrise în alfabetul A, care pot fi alcătuite prin aplicarea de un număr finit de ori a regulilor P1, P2 şi P3.

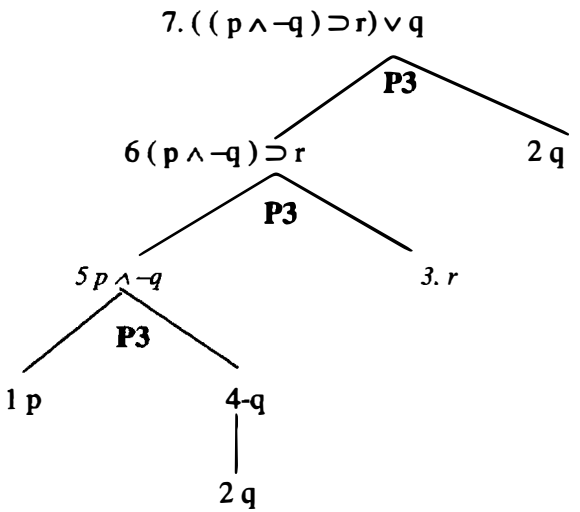
Exemple:

1. p	este formulă	conform regulii P1
2. q	este formulă	conform regulii P1
3. r	este formulă	conform regulii P1
4. $\neg$q	este formulă	conform regulii P2 (din 2)
5. (p $\wedge \neg$q)	este formulă	conform regulii P3 (din 1 şi din 4)
6. (p $\wedge \neg$q) $\supset$ r	este formulă	conform regulii P3 (din 5 şi 3)
7. ((p $\wedge \neg$q) $\supset$ r) $\vee$ q	este formulă	conform regulii P3 (din 6 şi 2)

Este uşor de observat că definiţia dată limbajului logicii propoziţiilor este de tip inductiv, constructivist. Ea ne permite să identificăm mulţimea formulelor bine formate după procedura prin care pot fi acestea obţinute. Tipul acesta de definiţii mai poate fi numit şi definiţii procedurale sau genetice, complet diferite de definiţiile semantic descriptive sau de defmiţiile prin gen proxim şi diferenţă specifică.

Exemplele date mai sus ilustrează înaintarea de la simplu la complex în construirea agregatelor propoziţionale. Putem să ne imaginăm şi opusul acestui demers, în care, fiindu–ne dat un agregat simbolic complex, ni se cere să determinăm, printr–un număr finit de paşi, dacă expresia dată satisface sau nu exigenţele limbajului logicii propoziţiilor. In acest caz ,vom reduce succesiv, din treaptă în treaptă, formula iniţială la subformulele ei imediate, şi acestea la rândul lor, la propriile lor subformule, până când accedem la formule elementare, respectiv la simboluri din A1.

Pe această cale, oricărui conectiv logic binar îi vom asocia două subformule constând din cele două argumente ale conectivului logic. În mod analog, unei formule prefixate de semnul negaţiei (ca un conectiv unar) îi vom asocia argumentul operatorului negaţiei. Transpunem, mai jos, analiza formulei 7, ultima din cele construite mai sus.



Arborele de mai sus poate fi numit *arbore de analiză a corectitudinii sintactice* a unei formule. Pe scurt, pentru a verifica corectitudinea sintactică a unei formule identificăm, mai întâi, operatorul ei principal şi plasăm formula dată în rădăcina unui arbore. Negaţiei îi asociem formula argument, iar conectorilor binari le asociem argumentele acestora. Procedeul se repetă până ce ajungem la variabile propoziţionale, respectiv, la propoziţii atomare, ce pot fi

adevărate sau false. Propoziţiile atomare vor eticheta nodurile terminale sau „frunzele" arborelui.

Este demn de reţinut faptul că putem defini doar prin trei reguli simple o mulţime infinită de formule bine formate ce alcătuieşte limbajul logicii propoziţiilor. În sfârşit, este util să reamintim că logica propoziţiilor nu ne îngăduie analiza internă a propoziţiilor, ce anume constante individuale sau variabile, semne funcţionale sau simboluri predicative intervin în structura acesteia. Din perspectiva logicii propoziţiilor, variabilele propoziţionale desemnează aserţiuni sau enunţuri ce pot fi adevărate sau false. Ea ne permite să studiem dependenţa valorii de adevăr a propoziţiilor complexe de valoarea de adevăr de propoziţiile elementare.

Regulile de formare prezentate mai sus sunt pur sintactice şi pot fi descrise în cadrul teoriei limbajelor formale.

Un limbaj logic formal este o structură posibilă, un cadru sau o formă în care putem turna conţinutul concret al enunţurilor noastre. Putem compara un limbaj formal cu diferite coşuri, vase, carafe, recipienţi în care noi putem turna diferite fructe, seminţe, băuturi, uleiuri. Dar aceste forme pot fi mereu combinate cu alte forme.

Ideea de bază este asocierea variabilelor propoziţionale cu propoziţiile declarative elementare, de exemplu, putem conveni să legăm variabila p cu propoziţia „plouă", propoziţia q cu propoziţia „bate grindina" şi variabila r cu propoziţia elementară „ creşte porumbul". Atunci formula compusă $(p \wedge \neg q) \supset r$ va descrie propoziţia condiţională sau clauza: „dacă plouă şi nu bate grindina, atunci creşte porumbul". Orice limbă naturală conţine conective logice ca „şi", „sau", „dacă..., atunci...", etc. Un limbaj logic formal este o structură elastică şi flexibilă care poate reda, schematic, abstract, anumite relaţii şi dependenţe intre clasele de obiecte, acte sau operaţii descrise în limbile naturale.

Limbajele logice formale permit cercetarea unor proprietăţi logice ale discursului uman cum sunt deducţiile sau inferenţele, interpretările sau modelele etc.

3. Operaţii şi funcţii logice. Evaluări semantice

Până acum nu am prezentat decât carcasa sintactică a logicii propoziţiilor şi am făcut abstracţie de conţinutul informaţional al propoziţiilor şi de valoarea de adevăr a acestora. Logica propoziţiilor

este o logică bivalentă. Ea aparţine teoriilor logice clasice care admit, în exclusivitate, ca valori de adevăr doar *adevărul* şi *falsul*. Convenim să notăm adevărul prin 1 şi falsul prin 0, în conformitate cu tradiţia instaurată de George Boole.

Negaţia, conjuncţia, disjuncţia implicaţia etc. sunt conective logice cu ajutorul cărora se alcătuiesc propoziţiile compuse.

Valoarea de adevăr a propoziţiilor compuse prin „şi", „sau", „dacă..., atunci..." depinde de valoarea de adevăr a propoziţiilor componente şi în ultimă instanţă a propoziţiilor elementare. Principiul consistenţei sau al noncontradicţiei ne obligă să atribuim unei propoziţii elementare, fie adevărul, fie falsul. In logica propoziţiilor, propoziţiilor elementare li se atribuie din oficiu valori de adevăr. Mai mult, sunt cercetate toate combinaţiile de valori posibile şi se arată pentru fiecare conectiv logic (conjuncţie, disjuncţie, implicaţie) pentru ce atribuiri de valori devine adevărat.

Conjuncţia este adevărată dacă ambele ei argumente(sau propoziţii conectate) sunt adevărate şi e falsă dacă vreunul dintre acestea este fals.

Disjuncţia este adevărată dacă vreunul dintre argumente este adevărat şi este falsă dacă ambele argumente sunt false.

Conectivele logice conjuncţia, disjuncţia, implicaţia, etc., vor putea fi definite ca funcţii booleene definite pe mulţimea B^n, unde $B = [0,1]$, cu valori în B. B^n descrie un şir de n atribuiri de valori din 0 sau 1 date celor n variabile propoziţionale. v este funcţia de valorizare:

$$v : B^n \longrightarrow B$$

Putem acum defini inductiv valoarea de adevăr a unei formule bine formate în limbajul L al logicii propoziţiilor printr–un număr de reguli semantice.

1. Dacă $\alpha \in A1$, respectiv α este o variabilă propoziţională, atunci, potrivit principiului bivalenţei, $v(\alpha) = 1$ sau $v(\alpha) = 0$.

2. Dacă $\alpha = -\beta$, atunci $v(\alpha) = 1$, { dacă, $\beta = 0$
   $v(\alpha) = 0$, { dacă, $\beta = 1$;

3. Dacă $\alpha = (\beta \wedge \gamma)$, atunci $v(\alpha) = 1$, { dacă, $v(\beta) = v(\gamma) = 1$;
   altfel $v(\alpha) = 0$;

4. Dacă $\alpha = (\beta \vee \gamma)$, atunci $v(\alpha) = 0$, { dacă $v(\beta) = v(\gamma) = 0$;
   altfel $v(\alpha) = 1$;

5. Dacă $\alpha = (\beta \supset \gamma)$, atunci $v(\alpha) = 0$ $\begin{cases} \text{dacă } v(\beta) = 1 \text{ şi } v(\gamma) = 0 \\ \text{altfel } v(\alpha) = 1; \end{cases}$

6. Dacă $\alpha = (\beta \equiv \gamma)$, atunci $v(\alpha) = 1$ $\begin{cases} \text{dacă, } v(\beta) = v(\gamma); \\ \text{altfel } v(\alpha) = 0. \end{cases}$

Pe calea aceasta am definit valoarea de adevăr a oricărei formule $\alpha \in L$. Strategia urmată este următoarea.

1. Orice propoziţie elementară este adevărată sau falsă, (principiul bivalenţei).

2. Negaţia unei propoziţii are întotdeauna valoarea de adevăr opusă propoziţiei iniţiale.

3. Conjuncţia este adevărată, dacă sunt adevărate ambele propoziţii componente; altfel e falsă.

4. Disjuncţia este falsă, dacă sunt false ambele propoziţii componente; astfel este adevărată.

5. Implicaţia este falsă, dacă este adevărată propoziţia antecedent şi falsă propoziţia consecvent; altfel este adevărată.

6. Echivalenţa este adevărată, dacă cele două propoziţii au aceeaşi valoare de adevăr; altfel e falsă.

Cele şase reguli alcătuiesc *un procedeu semantic inductiv* pentru calcularea valorii de adevăr a *oricărei* propoziţii compuse. Avem de a face cu o inducţie de la simplu la complex, de la propoziţii „atomice" la propoziţii moleculare formate în fiecare moment cu ajutorul uneia dintre regulile **2–6.** Prima regulă garantează *iniţializarea* procedeului inductiv, celelalte garantează continuarea acestuia până la determinarea valorii de adevăr a formulei complexe în cauză.

Aceste reguli definesc *semantica logicii propoziţiilor.*

Regulile **2–6** pot fi redate şi sub forma unor tabele de adevăr. Admitem, ca, în regula P1, că α şi β sunt „atomi" şi deci, vor lua, fiecare dintre ele, valorile 1 sau 0. Fiind date n variabile propoziţionale (propoziţii elementare sau atomi), vor fi 2^n atribuiri de valori binare. La începutul tabelului, în prima linie, ataşăm fiecărei variabile câte o coloană, apoi fiecărei operaţii logice câte o coloană distinctă.

Obţinem astfel tabloul nr 1:

Tabloul 1 - Evaluarea principalelor operaţii logic–propoziţionale

v (β)	v (γ)	v(-β)	v (β ∧γ)	v (β ∨γ)	v (β ⊃γ)	v (β≡γ)
1	1	0	1	1	1	1
1	0	0	0	1	0	0
0	1	1	0	1	1	0
0	0	1	0	0	1	1

În tabloul nr.1 sunt definite, în ordine, negaţia (coloana treia), conjuncţia, disjuncţia, implicaţia şi echivalenţa (coloanele 4–7) ca operaţii sau funcţii logice de unul sau două argumente β şi γ.

4. Operaţiile logice şi limba naturală

Înainte de a trece la analiza exhaustivă a funcţiilor logice de două argumente este util să ne oprim puţin asupra celor cinci operaţii şi corespunzător funcţiilor logice prezentate în tabloul nr.1 şi asupra rolului lor în limba naturală şi în gândirea logică.

Negaţia este o funcţie unară. Ea asociază fiecărei propoziţii având o valoare de adevăr determinată, valoarea de adevăr opusă. Prin negaţie, deci, se schimbă valoarea de adevăr a unei propoziţii. Negaţia este un *inversor* al valorii de adevăr a formulei iniţiale.

Se impun câteva distincţii. Mai întâi, operaţiile logice nu trebuie confundate cu rezultatul lor notat prin acelaşi simbol. O operaţie unară sau monadică asociază unui semn propoziţional sau operand un alt semn propoziţional. La fel, o operaţie binară asociază unei perechi de semne propoziţionale sau unui cuplu de operanzi un nou semn propoziţional, rezultatul operaţiei. Evident, operanzii sau argumentele sunt evaluate în domeniul de valori $V = B = \{0, 1\}$ şi în acelaşi domeniu vor fi evaluate şi rezultatele operaţiilor.

Atât operaţia monadică, negaţia, cît şi cele binare sunt iterabile. În consecinţă, o propoziţie de formă negativă poate fi, ea însăşi, negată, ajungându–se la o dublă negaţie.

În limbile naturale *negaţia poate fi realizată prin procedee stilistice diferite.*

Fie enunţul afirmativ:

4.1. Exerciţiul este corect.

Putem să–l negăm cel puţin în următoarele patru feluri:

4.2. Exerciţiul ***nu*** este corect.

4.3. Exerciţiul este *in*corect.
4.4. *Nu este adevărat* că exerciţiul este corect.
4.5. *Este fals* că exerciţiul este corect.

Din punctul de vedere al logicii propoziţiilor expresiile 4.2 – 4.5 sunt echivalente: fiecare dintre ele va avea o *valoare de adevăr opusă* propoziţiei iniţiale 4.1. şi, în consecinţă, va avea aceeaşi valoare de adevăr.

Pentru cercetarea mijloacelor logice ale limbilor naturale va fi totuşi util să distingem între cazurile 4.2 – 4.5.

Mai întâi propoziţia iniţială 4.1 descrie ceea ce în logica aristotelică se numea o judecată de predicţie având structura „S este P" sau „S ∈ P" (S = subiectul, P = predicatul şi „este" = copula. Vom numi acest fel de *negaţie, negaţie copulativă.* În cazul ei se trece de la S ∈ P la S ∉ P.

În 4.3 particula negatoare „in" se fixează asupra predicatului, schimbând pe această cale valoare de adevăr a propoziţiei. Vom numi această modalitate de a realiza negaţia drept *negaţie predicativă.* În cazul ei se trece de la S ∈ P la S ∈ -P.

În 4.4 şi 4.5 actul negaţiei se realizează prin construirea la nivel „metateoretic" a unei propoziţii despre propoziţia iniţială. Vom numi acest procedeu de a realiza negaţia drept *negaţia prefixată* sau *metateoretică.* Într–o logică bivalentă clasică prefixele „Nu este adevărat că" şi „Este fals că", ce apar în 4.4 şi 4.5 joacă roluri identice.

La nivelul logicii propoziţiilor nu sunt luate în consideraţie aceste subtilităţi. Dar în teoria operaţiilor cu judecăţi predicative dezvoltată de Aristotel astfel de distincţii întemeiau, de exemplu, validitatea inferenţei prin *obversiune.* Propoziţia de la 4.1 este echivalentă ca valoare de adevăr cu

4.6. Exerciţiul *nu* este *in*corect,

obţinută din 3.1 prin obversiune.

Obversiunea este, în esenţă, o lege a dublei negaţii pentru enunţurile asertorice de predicţie întemeiată pe *anihilarea unei negaţii copulative cu o negaţie predicativă.* Demersul obţinerii lui 4.6 este trecerea de la o propoziţie afirmativă la una cu negaţie predicativă şi de la aceasta la una cu negaţie copulativă (4.1 → 4.3 → 4.6).

Logica propoziţiilor centrată pe valorile de adevăr nu distinge între 4.1 şi 4.6. O teorie logică mai subtilă, care şi-ar propune să surprindă interacţiunea dintre subiecţii logici într-un *dialog,* ar putea distinge între 4.1, 4.3 şi 4.6 ca reprezentând etape distincte într-un dialog: afirmarea, contestarea, contestarea contestării:

4.1. $\models_x S \in P$ afirmaţie

4.3. $\models_y S \in -P$ contestaţie

4.6. $\models_x S \notin -P$ contestarea contestaţiei

4.4. $\models_x v(S \in -P) \neq 1$ contestarea contestaţiei metateoretică.

Dar astfel de obiective transcend limitele logicii propoziţiilor. Le-am menţionat doar întrucât sunt intim legate de operaţiile logice şi au făcut cândva obiectul preocupărilor logicii. Ele pot fi utile în edificarea unei teorii moderne a argumentării.

Conjuncţia este exprimată în limbile naturale prin particule ca „şi", „iar", „dar", prin virgule etc. Legarea prin conjuncţie a două propoziţii nu instituie o relaţie de ordine temporală sau valorică, ci doar o întărire sau restrângere a condiţiilor de adevăr:

Propoziţiile:

4.7. Se înserează

4.8. Se răceşte vremea

pot fi unite prin conjuncţie în:

4.9. Se înserează şi se răceşte vremea.

Ultima propoziţie va fi adevărată, dacă şi numai dacă este adevărat că „se înserează" şi este adevărat că „se răceşte vremea". Falsitatea uneia dintre propoziţiile componente impune falsitatea propoziţiei compuse şi numai adevărul tuturor membrilor unei conjuncţii asigură adevărul conjuncţiei.

Conjuncţia ca operaţie logică este insensibilă la relaţia de ordine. Fie p = „a murit" şi q = „l-a îngropat". $v(p \wedge q) = 1 \Leftrightarrow v(p) = 1$ şi $v(q) = 1$.

Propoziţia din limba naturală descrie o relaţie de succesiune necomutativă între moartea şi îngroparea unei persoane, ordinea celor două evenimente nefiind inversabilă.

Dimpotrivă, conjuncţia logic propoziţională descrie o relaţie atemporală, neordonată cauzal sau temporal, care satisface relaţia de comutativitate.

Conjuncţia logic matematică este o funcţie de adevăr şi nu poate descrie satisfăcător naraţiunile, planurile, dialogurile, care presupun relaţia de succesiune.

Avem nevoie de o altă specie de conjuncţie capabilă să descrie naraţiuni, programe, planuri sau execuţii de acte umane care să asimileze relaţia de ordine a argumentelor şi în care legea comutativităţii încetează din punct de vedere semantic. Într-o astfel de teorie logică nu va mai fi adevărată formula:

$$v(p \wedge q) = v(q \wedge p),$$

căci semnificaţia propoziţiei compuse „a murit şi l-a îngropat" este substanţial diferită de semnificaţia propoziţiei compuse „l-a îngropat şi a murit".

În cazul legării prin *disjuncţie* a două sau mai multe propoziţii este suficient să fie una singură adevărată pentru ca întregul agregat să fie adevărat.

4.10. Propoziţia „Ionescu Gheorghe este student la Automatică sau laureat al premiului Oscar pentru cinematografie sau campion mondial la triplu salt sau..." este adevărată, dacă este adevărată:

4.11. propoziţia „Ionescu Gheorghe este student la Automatică", chiar dacă nici una dintre celelalte alternative nu este satisfăcută.

Dimpotrivă, pentru a atesta falsitatea unei propoziţii disjunctive trebuie să probăm falsitatea *fiecăreia* dintre propoziţiile componente.

Disjuncţia inclusivă ilustrată mai sus nu trebuie confundată cu *disjuncţia exclusivă*, chiar dacă în limba naturală este redată prin aceeaşi particulă „sau":

4.12. „Ion este celibatar sau Ion este căsătorit"

este esenţial deosebită de:

4.13. „Petru este blond sau Petru este înalt"

căci în cazul 4.12 admiterea unei alternative exclude pe cealaltă, în timp ce în cazul 4.13 cele două alternative sunt reciproc compatibile. Petru poate fi, în acelaşi timp, blond şi înalt (4.13) în timp ce Ion nu poate fi în acelaşi timp celibatar şi căsătorit. Dacă este adevărat că Ion este celibatar, atunci, cu certitudine, Ion nu este căsătorit şi invers. Cele două atribute sunt reciproc incompatibile: una implică negaţia celeilalte.

Dintre toate operaţiile logice implicaţia este cea mai îndepărtată de experienţa bunului simţ şi de spiritul limbilor naturale.

Implicaţia logică postulează o dependenţă între valorile de adevăr a două propoziţii, şi anume, dacă propoziţia antecedent este adevărată, atunci va fi adevărată şi propoziţia consecvent.

4.14. Dacă triunghiul este isoscel, atunci are două unghiuri egale.

Implicaţia spune că nu este posibil să fie adevărată propoziţia antecedent, fără să fie, în acelaşi timp, adevărată propoziţia consecvent.

Probarea falsităţii unei implicaţii se reduce la găsirea unui contraexemplu, respectiv a unui caz în care antecedentul implicaţiei să fie adevărat, iar consecventul ei fals.

Echivalenţa poate fi caracterizată ca o implicaţie bilaterală. Ea se exprimă în limba naturală prin sintagma „dacă şi numai dacă". Negaţia echivalenţei este disjuncţia exclusivă.

Încheiem aici observaţiile noastre despre redarea operaţiilor logice în limbile naturale reţinând următoarele:

Negaţia se exprimă prin „nu", „in", „a–", sau prin construcţii valorizatoare ca „nu este adevărat că"; conjuncţia prin „şi", „dar", „iar", fără însă ca acestea să epuizeze posibilităţile de redare a operatorului logic în limbile naturale. În anumite situaţii acţionale, în exprimarea ironică, tonul poate servi ca un operator de infirmare. La fel „sau" este adesea ambiguu în limba naturală, exprimând, deopotrivă, disjuncţia inclusivă şi pe cea exclusivă. Limbile naturale nu sunt construcţii logice transparente, lipsite de ambiguităţi şi situaţii paradoxale. Aceasta este una dintre cauzele care au condus, cu ani în urmă, la eşuarea proiectelor de traducere automată dintr–o limbă naturală în altă limbă naturală. Pentru a formaliza corect un fragment sau altul din limba naturală aceasta are nevoie de un proces de dezambiguizare sau refacere a structurilor semantice şi sintactice. Limbile naturale sunt legate adesea de context şi situaţii pragmatice pentru care teoria logică clasică nu dispune de instrumente. De abia unele sisteme logice neclasice de factură modală, temporală, epistemică, acţională, etc. dau seama de o serie de alţi operatori logici existenţi în limbile naturale.

Am vorbit în capitolul introductiv despre cuantificatorii logici şi tipologia cuantificatorilor în limbile naturale. Nu vom relua aici observaţiile făcute acolo. O singură ideie trebuie şi aici relevată. Limbile naturale dispun de o gamă largă de cuantificatori încă netranspuşi în teoria logică.

Teoria deducţiei, pe de o parte, şi teoria argumentării, pe de altă parte, ne conduc şi ele la semnalarea unor particule lingvistice cu semnificaţie logică aparte. Voi atrage atenţia asupra particulelor „deci", sau „ prin urmare", care anunţă relaţia de consecinţă logică, şi asupra particulelor „deoarece", „căci", „întrucât", care anunţă trecerea de la o teză argumentată la temeiurile ei. Este util să reţinem că operatorul „deoarece" este inversul relaţiei de consecinţă logică.

Prima concluzie care se impune la sfârşitul acestui paragraf este una simplă şi stimulativă pentru cercetarea logică. Limbile naturale conţin, în stare latentă, sugestii şi impulsuri pentru elaborarea unor noi capitole de logică matematică, relevante pentru discursul uman şi pentru activităţile umane practice. Acestea se vor putea numi logici cu cuantificatori temporali, logici ale argumentării, logici ale acţiunii, teorii logice ale atitudinilor propoziţionale de acceptare, respingere sau îndoială, teoria logică a lui „sper că" şi „mă tem că", etc.

Cap. 2. TEORIA FUNCŢIILOR DE ADEVĂR

1. Formule şi funcţii logice. Tabloul funcţiilor logice bivalente

În paragraful 3 am interpretat operaţiile sau conectivele logice ca funcţii de adevăr, asociind fiecărei formule bine formate în logica propoziţiilor o funcţie logică definită pe domeniul B^n cu valori în B. Aceasta nu exclude însă posibilitatea ca aceiaşi funcţie de adevăr să fie exprimată prin formule deosebite, în care apar operatori logici deosebiţi. Am caracterizat în paragraful 3 şi 4 operaţiile logice principale negaţia, conjuncţia, disjuncţia, implicaţia şi echivalenţa şi am definit o regulă generală de evaluare a unei formule $\alpha \in L$. Este util să ne întrebăm dacă există, în afară de operaţiile logice analizate în paragrafele 3 şi 4, alte operaţii sau funcţii logice şi, în genere, care este numărul funcţiilor logice ce pot fi construite într-o logică cu m valori şi n argumente. Fiind date m valori de adevăr, i.e. $|V| = m$, şi n argumente pentru operaţiile sau conectivele logice, putem construi $(m^m)^n$ = funcţii logice m–valente de n argumente. În particular, într–o logică bivalentă i.e. m = 2 şi cu operaţii binare şi respectiv funcţii de două argumente n = 2, putem construi $(2^2)^2 = 4^2 = 16$ funcţii bivalente. După cum am văzut (3), aceste funcţii sunt definite pe domeniul B^n cu valori în B = [0,1]. Cum n = 2, ele vor fi definite pe domeniul [00, 01, 10, 00] cu valori în [0,1]. Putem să reprezentăm aceste funcţii în tabloul nr. 2.

	p	q	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
1	0	1	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
2	1	0	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
3	1	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
			C	$\wedge$	L	p	J	q	$\oplus$	$\vee$	$\downarrow$	$\equiv$	-q	$\subset$	-p	$\supset$	/	T

Tabloul 1 - Funcţiile logice bivalente

În partea stângă a tabloului de la 0 la 3 sunt ordonate cele 4 atribuiri de valori posibile pentru două variabile, p şi q. Numărul

zecimal al atribuirii coincide cu numărul binar format de combinaţiile de zero şi unu şi el poate fi calculat după formula:

$$\alpha = a_1 * 2^{n-1} + a_2 * 2^{n-2} + ... + a_{n-1} * 2 + a_n$$

Astfel cea de a treia atribuire va avea numărul $= 1*2 + 0 = 2$.

Pe prima linie a tabloului sunt numerotate funcţiile logice bivalente, iar pe ultima linie este scris simbolul funcţiei logice în cauză. Funcţia nr. 0 este inconsistenţa sau irealizabilitatea pe care convenim să o notăm prin C. Funcţia nr.1 este conjuncţia. Vom avea apoi, în ordine, excepţia sau contraimplicaţia, asertarea lui p, excepţia directă sau negaţia implicaţiei directe, asertarea lui q, disjuncţia exclusivă şi disjuncţia inclusivă.

Funcţiile care urmează după disjuncţia inclusivă vor fi negaţii ale funcţiilor enumerate până acum. Astfel funcţia 8 „nici" este negaţia lui 7, disjuncţia inclusivă; 9, echivalenţa este negaţia lui 6, disjuncţia exclusivă şi aşa mai departe.

Putem institui următoarea regulă:

(R 5.1) Pentru orice m şi n, dacă $1 \leq m, n \leq 15$ şi $m + n = 15$, vom avea $f_n = - f_m$.

De exemplu, pentru $m = 6$, $n = 9$, cum $6+9=15$, vom avea $f_9 = - f_6$,respectiv echivalenţa logică este non disjuncţie exclusivă (vezi mai jos 5.6).

Regula instituită ne permite, între altele, să scriem următoarele echivalenţe logice:

5.1 $T \equiv -C$

5.2 $(p/q) \equiv -(p \wedge q)$

5.3 $p \llcorner q) \equiv -(p \supset q)$

5.4 $--p \equiv p$

5.5 $(p \subset q) \equiv -(p \llcorner q)$

5.6 $(p \equiv q) \equiv -(p \oplus q)$

5.7 $(p \downarrow q) \equiv -(p \vee q)$

Tautologia sau funcţia identic adevărată este negaţia irealizabilităţii (5.1).

Incompatibilitatea este negaţia conjuncţiei (5.2).

Excepţia este negaţia implicaţiei (5.3).

Negarea negaţiei este echivalentă cu propoziţia iniţială (5.4)

Implicaţia inversă este echivalentă cu negaţia excepţiei inverse (5.5)

Echivalenţa este nondisjuncţie exclusivă (5.6).
„Nici" este nondisjuncţie inclusivă (5.7).

Tautologiile sau funcţiile identic–adevărate sunt importante în teoria logică pentru că descriu legi logice.

Cele 16 funcţii bivalente pot fi, între altele, clasificate după numărul atribuirilor în care iau valoarea 1.

I. Funcţia nr.0 nu are nici o atribuire de 1.

II. Funcţiile nr.1, 2, 4 şi 8 au câte o singură atribuire de valori în care sunt adevărate.

III. Funcţiile nr.3, 5, 6, 9, 10, 12 au câte două atribuiri de valori în care devin adevărate.

IV. Funcţiile nr.7, 11, 13, 14 au câte trei atribuiri de valori în care devin adevărate.

V. Funcţia nr.15 are patru atribuiri de valori în care devine adevărată.

Clasificarea funcţiilor de adevăr după numărul atribuirilor în care devin adevărate ne va permite, după ce vom introduce conceptele de implicant şi consecinţă logică, să formulăm nişte reguli de scriere a unor legi logice direct din observarea tabloului nr. 2.

2. Tabloul funcţiilor bivalente şi teoria mulţimilor

Să ne imaginăm pentru moment că variabilele p şi q desemnează două propoziţii elementare şi că mulţimea W desemnează mulţimea situaţiilor în care p şi q pot fi adevărate sau false. Admitem, de asemenea, că funcţia de valorizare v este definită acum pe $A_1 \times W$ cu valori în $B = \{0,1\}$. (A_1 este alfabetul de variabile propoziţionale).

Putem atunci să asociem fiecărei propoziţii elementare o submulţime din W în care propoziţia în cauză devine adevărată. Submulţimi analoge vor fi asociate şi negaţiilor lui p şi q.

$$A = \{ w \in W \mid v(p, w) = 1 \}$$

$$B = \{ w \in W \mid v(q, w) = 1 \}$$

A şi B vor fi mulţimile caracteristice ale propoziţiilor elementare p şi q. Admiţând că p şi q pot fi ambele adevărate, adevărată una fără să fie şi cealaltă şi ambele false, obţinem reprezentarea pe W a celor patru atribuiri de valori descrise în tabelul 2: ambele false, adevărată numai q, adevărată numai p, ambele adevărate (fig.2.1).

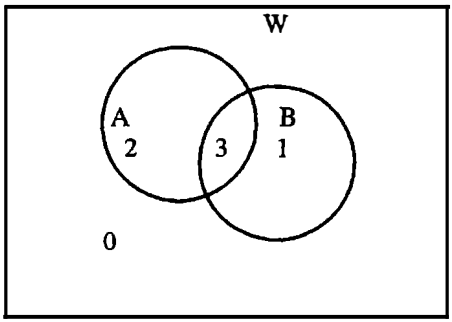


Fig 2 1. Reprezentarea asamblistă a funcţiilor bivalente

Pe această cale, mulţimea W a fost partiţionată în patru submulţimi disjuncte corespunzătoare celor patru atribuiri de valori.

$0 = -A \cap -B$	$-p \wedge -q$	$p = q = 0$
$1 = -A \cap B$	$-p \wedge q$	$p = 0, q = 1$
$2 = A \cap -B$	$p \wedge -q$	$p = 1, q = 0$
$3 = A \cap B$	$p \wedge q$	$p = q = 1$

Să considerăm acum mulţimea părţilor lui *W*, 2^W Mulţimea vidă $\varnothing$ aparţine oricărei mulţimi; de aceea o vom considera prima submulţime a lui *W*.

W I $\{\varnothing\}$

W II $\{0\}, \{1\}, \{2\}, \{3\}$

W III $\{0, 1\}, \{0, 2\}, \{0, 3\}, \{1, 2\}, \{1, 3\}, \{2, 3\}$

W IV $\{0, 1, 2\}, \{0, 1, 3\}, \{0, 2, 3\}, \{1, 2, 3\}$

W V $\{0, 1, 2, 3\}$

Numărul submulţimii părţilor lui W = $\{0, 1, 2, 3\}$ este dat de

$C^0_n + C^1_n + C^2_n + C^3_n + C^4_n = 16$

Este uşor de observat că familia de mulţimi W II este de bază: toate celelalte familii nevide fiind construite cu elemente din W II.

Să ne întoarcem acum la clasificarea celor 16 funcţii bivalente. Este uşor să observăm o corespondenţă între clasificarea funcţiilor bivalente după numărul lor de adevăr (vezi 5) şi cele cinci familii de submulţimi W I – W V enumerate mai sus.

W I $\varnothing = \neg \{0, 1, 2, 3\}$ funcţia nr. 0 în tabelul nr. 2

W II

Tabloul 2 – Reprezentarea funcţiilor logice cu un singur caz de adevăr

{0}	{1}	{2}	{3}
$-A \cap -B$	$-A \cap B$	$A \cap -B$	$A \cap B$
$-p \wedge -q$	$-p \wedge q$	$p \wedge -q$	$p \wedge q$
$p \downarrow q$	$p \rfloor q$	$p \lfloor q$	$p \wedge q$
8	4	2	1

W III

Tabloul 4 Reprezentarea funcţiilor logice cu două cazuri de adevăr

{0,1}	{0,2}	{0,3}	{1,2}	{1,3}	{2,3}
-A	-B	$-A \cap -B \cup A \cap B$	$-A \cap B \cup A \cap -B$	B	A
-p	-q	$-p \wedge -q \vee p \wedge q$	$-p \wedge q \vee p \wedge -q$	q	p
-p	-q	$p \equiv q$	$p \oplus q$	q	p
12	10	9	6	5	3

W IV

Tabloul 5 Reprezentarea funcţiilor logice cu câte trei cazuri de adevăr

{0,1,2}	{0,1,3}	{0,2,3}	{1,2,3}
$-A \cap -B \cup -A \cap B \cup A \cap -B$	$-A \cap -B \cup -A \cap B \cup A \cap B$	$-A \cap -B \cup A \cap -B \cup A \cap B$	$-A \cap B \cup A \cap -B \cup A \cap B$
$-A \cup A \cap -B$	$B \cup -A \cap -B$	$-B \cup A \cap B$	$B \cup A \cap -B$
$-A \cup -B$	$-A \cup B$	$A \cup -B$	$A \cup B$
$-p \vee -q$	$-p \vee q$	$p \vee -q$	$p \vee q$
p / q	$p \supset q$	$p \subset q$	$p \vee q$
14	13	11	7

Tabloul 6. Reprezentarea grafică a funcţiei cu patru cazuri de adevăr

{0,1,2,3}
-A∩-B ∪ -A∩B ∪ A∩-B ∪ A∩B
-A ∪A
W
15

Toate cele 16 funcţii de adevăr au reprezentări asambliste. Funcţiile incluse în clasa II în clasificarea după numărul cazurilor de adevăr ce coincid cu cele 4 atribuiri de valori enumerate la W II stau la baza definirii celorlalte clase de funcţii III, IV, şi V;

Cele patru funcţii adevărate într–un singur caz de adevăr pot fi caracterizate în exclusivitate prin conjuncţie şi negaţie (vezi W II rândul 3), iar dacă la acestea adăugăm disjuncţia, atunci obţinem o algebră booleană la care putem reduce toate celelalte funcţii de adevăr.

Există un izomorfism între funcţiile bivalente şi mulţimea părţilor unei mulţimi alcătuite cu mulţimile caracteristice asociate variabilelor propoziţionale.

Elementelor definitorii ale logicii propoziţiilor le sunt puse în corespondenţă elemente fundamentale din algebra mulţimilor. Pe această cale obţinem o semantică reprezentată într-un plan al logicii propoziţiilor.

Tabloul 7. Logica propoziţiilor şi algebra mulţimilor

Logica propoziţiilor	Algebra mulţimilor
∧	∩
∨	∪
–	¬
⊃	⊂
a, f	1,0
L	W

Algebra mulţimilor şi logica propoziţiilor pot fi ambele descrise ca interpretări ale unor algebre booleene.

Proprietăţile formale ale logicii propoziţiilor pot fi descrise în cadrul unei algebre booleene definită de o mulţime cu două elemente şi cinci operaţii: două binare, una unară şi două nulare (constante).

$$B = \langle \{ 0, 1 \}, \wedge, \vee, ', 0, 1 \rangle$$

satisfăcând proprietăţile de comutativitate, asociativitate, distributivitate, margine universală şi complementaritate:

	s.	d.
L1	$x \wedge y = y \wedge x$	$x \vee y = y \vee x$
L2	$x \wedge (y \wedge z) = (x \wedge y) \wedge z$	$x \vee (y \vee z) = (x \vee y) \vee z$
L3	$x \wedge (y \vee z) = (x \wedge y) \vee (x \wedge z)$	$x \vee (y \wedge z) = (x \vee y) \wedge (x \vee z)$
L4	$x \wedge 0 = 0$	$x \vee 0 = x$
L5	$x \wedge 1 = x$	$x \vee 1 = 1$
L6	$x \wedge x' = 0$	$x \vee x' = 1$
L7	$(x')' = x$	

3. Principiul dualităţii

Dacă în teoremele algebrei booleene vom substitui între ele operaţiile $\wedge$ şi $\vee$ şi, de asemenea, constantele 1 şi 0 obţinem tot teoreme.

Aplicat asupra axiomelor L1 – L6 principiul dualităţii nu face decât să inverseze între ele cele două coloane atestând prin aceasta valabilitatea sa asupra axiomelor algebrei booleene.

Din axiomele L1 – L6 pot fi uşor demonstrate principiul idempotenţei, legea absorbţiei, legile lui De Morgan etc. Ilustrăm aceasta prin principiul idempotenţei şi legile absorbţiei.

T1. $x \vee x = x$

$x \vee x = (x \vee x) \wedge 1 = (x \vee x) \wedge (x \vee x') = x \vee (x \wedge x') = x \vee 0 = x$

S-au aplicat în ordine axiomele: L5s, L6d, L3d, L6s, L4d.
Aplicând principiul dualităţii la T1, obţinem:

T2 $x \wedge x = x$

T3. $x \vee (x \wedge y) = x$

$x \vee (x \wedge y) = (x \wedge 1) \vee (x \wedge y) = x \wedge (1 \vee y) = x \wedge 1 = x$

T3 este o lege de absorbţie.

S-au aplicat în ordine L5s, L3s, L5d, L5s şi s-au omis unele rezultate intermediare.

T4. $x \wedge (x \vee y) = x$
(PD1)

T4 este legea de absorbţie duală lui T3.

Principiul dualităţii este valabil şi în logica propoziţiilor. Axiomele L1 – L6 descriu legi şi în logica propoziţiilor.

Redăm mai jos câteva teoreme şi, corespunzător, câteva reguli privind utilizarea principiului dualităţii.

Fie $\alpha \in L$. Convenim să notăm duala lui α prin α^*.

DR1 Dacă α este o tautologie şi α^* este duala lui α, atunci $-(\alpha^*)$ este, de asemenea, tautologie.

DR2 Dacă $\alpha \supset \beta$ este o tautologie, atunci $\beta^* \supset \alpha^*$ este, de asemenea, tautologie.

DR3 Dacă $\alpha \equiv \beta$ este o tautologie, atunci $\alpha^* \equiv \beta^*$ este, de asemenea, o tautologie.

DR4 Dacă două formule α şi β au aceeaşi dual㠂, atunci sunt echivalente.

DR5 Dacă α este o tautologie, atunci duala sa este o contradicţie.

DR6 Duala dualei lui α este identică cu α , $((\alpha^*)^* = \alpha)$

DR7 Duala tabloului de adevăr al unei formule se obţine prin inversarea valorilor în atribuiri şi în coloana formulei.

Exemple:

Ex. DR1: $\alpha = -p \vee (p \vee q)$, $\alpha^* = -p \wedge (p \wedge q)$, $-(\alpha^*) = -(p \wedge q) \vee p$

Ex. DR2: Fie $\alpha = p$, $\beta = p \vee q$ şi $p \supset (p \vee q) = T$, atunci $(p \wedge q) \supset p = T$

Ex. DR3: Dacă $p \equiv p \wedge p = T$, atunci $p \equiv p \vee p = T$

Ex. DR4: Dacă $p \supset q$ şi $-p \vee q$ au ambele drept dual㠂 pe $-p \wedge q$, atunci ele sunt echivalente. În consecinţă, $(p \supset q) \equiv (-p \vee q)$.

Ex. DR5. Dacă $p \vee -p = T$, atunci $(p \vee -p)^* = p \wedge -p = C$

Ex. DR6: $\alpha = p \wedge -q$, $\alpha^* = p \vee -q$, $(\alpha^*)^* = p \wedge -q$

Ex. DR7: Fie $\alpha = -p \vee -q = p/q$ şi $\alpha^* = -p \wedge -q = p \downarrow q$ şi tablourile lor de valori:

pq	-p∨-q
00	1
01	1
10	1
11	0

pq	-p∧-q
11	0
10	0
01	0
00	1

Duala unei matrici logice se obţine prin inversarea valorilor de adevăr în domeniul atribuirilor de valori şi în coloanele funcţiilor logice.

4. Algebra funcţiilor de adevăr

În tabloul nr.2 din par. 5 am identificat cele 16 funcţii de adevăr bivalente. Luând mulţimea acestor funcţii drept mulţime iniţială F_2 şi drept operaţii binare produsul şi suma logică i.e. conjuncţia şi disjuncţia, iar ca operaţie unară complementaritatea, putem defini o algebră a funcţiilor de adevăr:

$$AF_2 = \langle F_2, ', \wedge, \vee, 1, 0 \rangle$$

unde orice $f \in F_2$ este definită pe mulţimea atribuirilor $\{00, 01, 10, 11\}$ cu valori în $\{0, 1\} = B$.

Definim mai întâi negaţia:

În orice punct $x \in B^2 = \{0, 1, 2, 3\} = \{00, 01, 10, 11\}$

$$8.1.\ f(x) = 1 - f(x)$$

i.e. negaţia valorii funcţiei f pentru o atribuire x va fi egală cu 1 minus valoarea funcţiei f în punctul x.

Analog, vom defini valorile funcţiilor produs (i.e. conjuncţie) şi respectiv sumă (disjuncţie).

Fie o funcţie produs $m = f \wedge g$ şi o funcţie sumă $h = f \vee g$. Atunci valoarea uneia dintre ele într–un punct $x \in B^2$ va fi dată de:

$$8.2.\ m(x) = \min(f(x), g(x))$$

pentru conjuncţie şi, respectiv, de:

$$8.3.\ h(x) = \max(f(x), g(x))$$

pentru disjuncţie.

Să considerăm, de exemplu, în tabloul 2 funcţiile f_1 şi f_7. Negaţiile lor vor fi, conform lui 8.1 funcţiile f_{14} şi f_8. Să considerăm pentru ilustrare funcţia f_1 şi atribuirile 0 la 2 pentru care avem $f_1(0) = f_1(1) = f_1(2) = 0$ şi în consecinţă $f'_1(0) = f'_1(1) = f'(2) = 1$. i.e., 1–0 = 1. Pentru $x = 3$ sau cea de a patra atribuire vom avea $f_1(3) = 1$ iar $f'_1(3) = 1 - 1 = 0$. Funcţia care are coloana de valori 1110 este într-adevăr funcţia f_{14}.

Similar se calculează negaţia lui f_7, obţinându-se f_8.

Să calculăm, acum, conjuncţia dintre f_{14} şi f_8, după formula 8.2, de mai sus. Vom parcurge comparativ coloanele 14 şi 8, asociate funcţiilor cu acelaşi număr, luând minimum pentru fiecare atribuire şi regăsind ca rezultat chiar coloana 8.

Definirea într-o atribuire $x \in B^2$ a unei funcţii $j = f \supset g$ se va face pe baza lui 8.1 şi 8.3, cunoscut fiind faptul că $f \supset g$ este reductibilă la $-f \vee g$.

Vom avea astfel:

$$8.4.\ j(x) = \max(1 - f(x), g(x))$$

Să calculăm:

$j(0) = f_1(0) \supset f_9(0) = \max(1 - f_1(0), f_9(0)) = \max(1, 1) = 1;$

$j(1) = f_1(1) \supset f_9(1) = \max(1, 0) = 1;$

$j(2) = (f_1(2) \supset f_9(2) = \max(1, 0) = 1;$

$j(3) = f_1(3) \supset f_9(3) = \max(0, 1) = 1.$

Rezultă, deci, că $j = f_1 \supset f_9$ este pentru orice atribuire $x \in B^2$, 1.

Dar $f_1 = p \wedge q$ şi $f_9 = p \equiv q$. Obţinem astfel $(p \wedge q) \supset (p \equiv q) = T$

Definiţiile 8.1–8.4 introduc operaţii algebrice funcţii de adevăr: complementaritatea, produsul, suma şi relaţia de ordonare. Acestea pot fi utilizate pentru identificarea legilor logice, pentru verificarea echivalenţei expresiilor etc.

Aşa, de exemplu, se pot verifica pe cale algebrică implicaţiile dintre f_1 şi f_3, dintre f_1 şi f_7, dintre f_1 şi f_{11} sau f_1 şi f_{13}

5. Interdefinibilitatea şi reducerea operaţiilor logice

Am văzut deja în capitolele **7** şi **8** că operaţiile logice şi corespunzător operaţiile algebrice booleene pot fi exprimate prin operaţiile lor duale. În consecinţă, pentru întemeierea logicii propoziţiilor nu este nevoie să acceptăm ca operaţii primitive toate operaţiile logice bivalente identificate în tabloul nr.2, ci este suficient să luăm ca primitive negaţia şi conjuncţia, negaţia şi disjuncţia sau negaţia şi implicaţia. Mai mult, putem edifica logica propoziţiilor acceptând ca operaţii primitive doar incompatibilitatea / sau operaţia „nici", celelalte operaţii urmând a fi introduse prin definiţie.

Cea mai des utilizată bază de concepte primitive pentru logica propoziţiilor este cea în care sunt admise ca primitive conectivele logic propoziţionale ce corespund operaţiilor booleene –, ∧ , ∨

Operaţii primare: –, ∧, ∨

Operaţii derivate: ⊃, ≡, /, ↓, ⊕, ⌋, ⌊

Definiţii

D1 $p \supset q = \neg p \vee q$

D2 $p \equiv q \ (p \supset q) \wedge (q \supset p)$

D3 $p/q = \neg(p \wedge q)$

D4 $p \downarrow q = \neg(p \vee q)$

D5 $p \oplus q = \neg(p \equiv q)$

D6 $p \lfloor q = \neg(p \supset q)$

D7 $p \rfloor q = \neg(q \supset p)$

Nu am mai introdus ca o funcţie distinctă implicaţia inversă ⊂, deoarece presupunând admisă ordinea alfabetică a variabilelor *p* şi *q*, am caracterizat–o prin q ⊃ p. Fiecare dintre definiţiile D2 – D7 pot fi aduse la forme în care apar exclusiv conective booleene. De exemplu, definitorul lui D2 poate fi înlocuit, pe baza lui D1, prin $(\neg p \vee q) \wedge (\neg q \vee p)$. La fel, definitorul lui D5 poate fi înlocuit succesiv prin $\neg((\neg p \vee q) \wedge (\neg q \vee p))$, $(\neg(p \vee q) \vee \neg(\neg q \vee p))$, $((p \wedge \neg q) \vee (q \wedge \neg p))$.

Întrucât operaţiile ∧ şi ∨ pot fi definite una prin alta cu ajutorul negaţiei, putem, evident, restrânge numărul operaţiilor primitive la – şi ∧ sau – şi ∨.

Primitive: –, ∧

Derivate: ∨ , ⊃, ≡, /, ↓, ⊕, ⌋, ⌊

D8 $p \vee q = \neg(\neg p \wedge \neg q)$

Pe baza lui D8 pot fi introduse D1, D2 şi toate celelalte definiţii de mai sus.

Putem lua ca primitive – şi ∨ :

Atunci introducem:

D9 $= p \wedge q = \neg(\neg p \vee \neg q)$

şi din nou obţinem conectivele booleene şi pe baza lor putem introduce, ca mai sus, D1 – D7.

Demn de menţionat este faptul că, atât în cazul lui D8, cît şi în cazul lui D9, am utilizat legea dublei negaţii introdusă la nivel de algebră booleană prin L7 (vezi cap. **5**). Exprimată logic–propoziţional aceasta este:

L7^0 $--p = p$

În sfârşit, putem lua drept primar un singur operator incompatibilitatea / sau pe „nici", i.e. $\downarrow$

În primul caz, definim negaţia ca incompatibilitate cu sine a propoziţiei în cauză:

D10 $-p = p/p$

şi conjuncţia şi disjuncţia ca:

D11 $p \wedge q = (p/q) / (p/q)$

D12 $p \vee q = (p/p) / (q/q)$

În cel de al doilea caz, definim negaţia prin „nici"

D13 $-p = p \downarrow p$

şi introducem analog conjuncţia şi disjuncţia ca:

D14 $p \wedge q = (p \downarrow p) \downarrow (q \downarrow q)$

D15 $p \vee q = (p \downarrow q) \downarrow (p \downarrow q)$

Toate celelalte conective binare pot fi introduse similar procedurii anterioare, când am admis ca primitive $-, \wedge$ şi $\vee$ (vezi D1–D7).

Cu cât un limbaj formal are un număr mai redus de concepte primare, cu atât definiţiile explicite vor fi mai complicate. Economia făcută la nivelul alfabetului se plăteşte cu preţul complicării procedurii de obţinere a conceptelor derivate.

6. Implicaţii şi consecinţe

Am văzut în par. 1 că $f_1 \supset f_9$, respectiv că funcţia conjuncţie implică funcţia echivalenţă şi că, analog, conjuncţia implică disjuncţia sau implicaţia etc. Suntem îndreptăţiţi să ne întrebăm ce este un implicant şi ce este o consecinţă a unei formule date.

D1. Fiind dată o funcţie f_i, vom numi *implicant* al ei orice funcţie f_k ce ia valoarea 0, ori de câte ori f_i ia valoarea 0.

D2 Fiind dată o funcţie f_i, vom numi *consecinţă* a ei orice funcţie f_k ce ia valoarea adevăr 1, ori de câte ori funcţia f_i ia valoarea adevăr 1.

Obs.1. Consecinţele conservă cazurile de adevăr ale unei funcţii, în timp ce antecedenţii sau implicanţii conservă cazurile de fals ale acesteia.

Obs.2. Atât determinarea implicanţilor cât şi determinarea consecinţelor permite identificarea de legi logice pornind de la o formulă dată.

Obs.3. Relaţia de consecinţă şi corespunzător de antecedent sunt una inversa celeilalte. Dacă q este consecinţa lui p, atunci p este

antecedentul lui *q*. Relaţia de consecinţă logică semantică se notează prin $\models$

$$A \models B \text{ sau } A_1,...,A_n \models B$$

Deoarece orice funcţie de adevăr este exprimată printr-o formulă, vom putea vorbi de relaţia de consecinţă logică şi de antecedenţi ai unei formule oarecare.

Se poate demonstra că dacă B este consecinţă logică din $A_1,...A_n$ i.e $A_1,...,A_n \models B$, atunci conjuncţia antecedenţilor implică consecinţa, i.e $A_1 \wedge \quad \wedge A_n \supset B = T$. Prin T am notat funcţia identic adevărată sau tautologia.

La fel, putem demonstra că, dacă $A_1 \wedge ... \wedge A_n \supset B = T$, atunci

$A_1 \wedge ... \wedge A_n \wedge -B = \square$

Aceasta revine la a afirma că propoziţiile:

I $A_1, ...,A_n \models B$

II $A_1 \wedge ... \wedge A_n \supset B = T$, unde prin T am notat tautologia.

III $A_1 \wedge ... \wedge A_n \wedge -B = \square$, unde prin $\square$ am notat contradicţia.

sunt echivalente.

Demonstrăm, mai întâi, prin reducere la absurd că I implică II.

1) $A_1,..A_n \models$ B ip.

2) II $A_1 \wedge ... \wedge A_n \supset B \neq T$

Propoziţia demonstraţiei indirecte 2) afirmă că există o atribuire de valori v (sau un model) în care:

3) $(A_1 \wedge ... \wedge A_n)^v = 1$

4) $B^v = 0$.

Potrivit definiţiei D2, propoziţia 1) poate fi rescrisă ca:

5) În orice interpretare *v*,

dacă $A^v_1 = A^v_2 =...= A^v_n = 1$, atunci $B^v = 1$.

Pe de altă parte din 3) şi definiţia conjuncţiei rezultă:

6) $A^v_1 = A^v_2 = ...=A^v_n = 1$.

Iar din 5) şi din 6) rezultă, prin *modus ponens*:

7) $B^v = 1$

ceea ce contrazice 4).

Prin aceasta s–a încheiat demonstraţia indirectă a faptului că I implică II.

Dăm acum demonstraţia inversă: II implică I.

Construim tot o demonstraţie prin reducere la absurd:

1) $A_1 \wedge ... \wedge A_n \supset B = T$ ip.

2) $A^v_1 = A^v_2 = ... = A^v_n = 1$ | ip. dem. indirecte

3) $B^v = 0$ |

Din adevărul lui 2) putem deduce, prin I $\wedge$, adevărul conjuncţiei:

4) $(A_1 \wedge A_2 \wedge ... \wedge A_n)^v = 1$.

Cum 1) este adevărată în orice interpretare, din 1) şi 4) rezultă:

5) $B^v = 1$

ceea ce contrazice pe 3). În consecinţă, demonstraţia se încheie.

Din cele două demonstraţii de mai sus rezultă echivalenţa dintre I şi II.

Să cercetăm acum relaţia dintre II şi III. Vrem să demonstrăm că II implică III şi invers.

Trecerea de la II la III se face prin negarea ambilor termeni ai unei egalităţi (de fapt a unei echivalenţe). Negaţia unei tautologii este irealizabilitatea sau contradicţia (a se vedea termenul din dreapta). Negaţia unei implicaţii este conjuncţia dintre afirmarea antecedentului şi contestarea consecventului (a se vedea termenul din stânga). III se obţine direct din II şi invers, prin proprietăţile relaţiei de echivalenţă.

Am demonstrat, deci, că I = II şi II = III. Întrucât echivalenţa este tranzitivă, I = III.

Vom vedea mai departe că aflarea antecedenţilor şi consecinţelor unei formule se poate obţine prin formele normale disjunctive şi corespunzător conjunctive şi că aceste procedee sunt esenţiale pentru verificarea validităţii raţionamentelor noastre. Acum vom întrebuinţa conceptele de implicanţi şi consecinţe pentru aprofundarea interpretării tabloului funcţiilor bivalente.

7 Pătratul logic şi tabloul funcţiilor bivalente

Logica tradiţională cercetează judecăţile de predicţie de forma S este P, prescurtat S ∈ P. S desemnează subiectul, P predicatul şi ∈ copula. După calitate, judecăţile erau împărţite în afirmative şi în negative. După

cantitate, în universale şi particulare. După ambele criterii se obţin judecăţi universal afirmative sau de tip A, universal negative sau de tip E, particular afirmative sau de tip I şi particular negative sau de tip O.

Relaţiile dintre aceste patru tipuri de judecăţi au fost sistematizate ulterior de către Apuleius în aşa numitul pătrat logic:

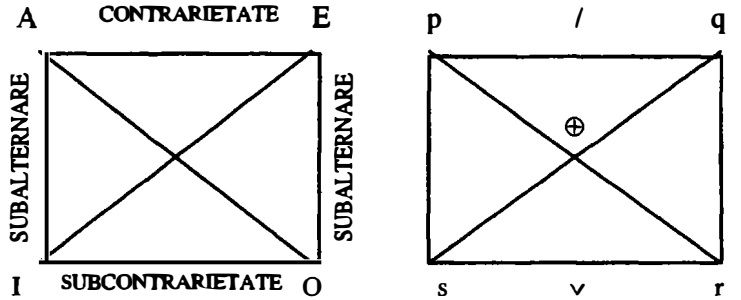


Fig 7. 1 Interpretări ale pătratului logic în logica clasică şi în logica propoziţiilor

Două judecăţi sunt în raporturi de contrarietate, dacă nu pot fi simultan adevărate. La fel sunt în raporturi de *subcontrarietate,* dacă nu pot fi simultan false. În sfârşit, sunt în raporturi de *contradicţie,* dacă nu pot fi, simultan, nici adevărate nici false, respectiv, sunt întotdeauna de valori de adevăr opuse. Două judecăţi sunt în raporturi de *subordonare,* dacă adevărul uneia presupune în mod necesar adevărul celeilalte.

Conceptul clasic de pătrat logic este pasibil de generalizare.

D1 Numim *pătrat logic* o funcţie logic propoziţională de patru variabile $\varphi(p, q, r, s)$ care satisface condiţiile:

1) p/q 3) $p \supset s$ 5) $p + r$

2) $s \vee r$ 4) $q \supset r$ 6) $q + s$

sau altfel scris:

$$\varphi(p, q, r, s) = (p/q) \wedge (s \vee r) \wedge (p \supset s) \wedge (q \supset r) \wedge (p + r) \wedge (q + s)$$

Este uşor de văzut că relaţia de contrarietate se descrie prin incompatibilitate, cea de subcontrarietate prin disjuncţia inclusivă, cea de contradicţie prin disjuncţia exclusivă şi cea de subalternare prin implicaţie (vezi fig. 7.1).

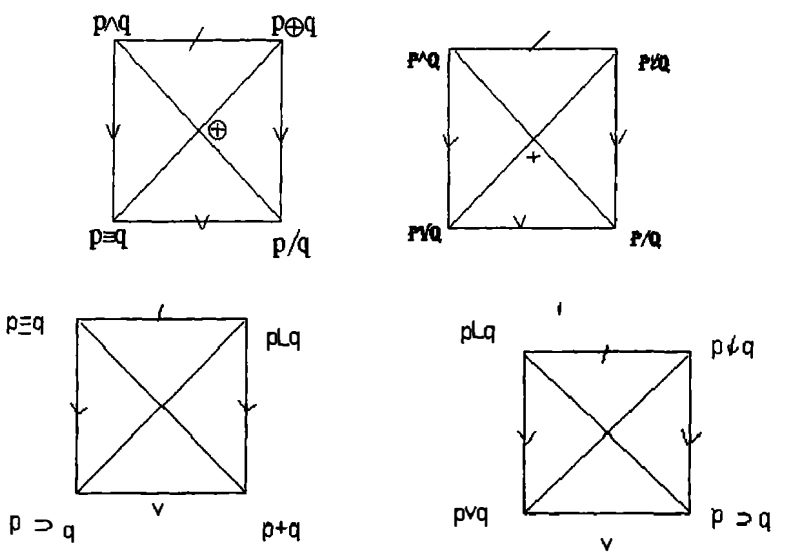
Utilizând funcţia pătrat logic φ (p, q, r, s), tabloul funcţiilor logice bivalente şi definiţia conceptului de consecinţă logică putem construi diferite interpretări ale pătratului logic în logica propoziţiilor. Pentru aceasta procedăm în felul următor:

1. Selectăm o funcţie adevărată într–un singur caz sau două cazuri de adevăr pe care o situăm în colţul *A* (respectiv *p)* al pătratului logic.
2. În colţul I (respectiv s) vom plasa o consecinţă logică a funcţiei alese.
3. În colţurile opuse lui A şi lui I (respectiv lui p şi lui s vom plasa negaţiile funcţiilor în cauză în conformitate cu regula R 5.1 din par. 5.

Interpretarea astfel obţinută satisface cele 16 inferenţe imediate specifice pătratului logic.

Fiecare dintre cele patru interpretări considerate evidenţiază un număr de 16 legi logice.

Exemple



Interpretări ale pătratului logic în logica propoziţiilor

Scriem legile logice evidențiate de prima interpretare:

1. $p \wedge q \supset (p \equiv q)$
2. $p \oplus q \supset (p/q)$
3. $(p \wedge q) / (p \oplus q)$
4. $(p \equiv q) \vee (p/q)$
5. $(p \wedge q) \equiv -(p/q)$
6. $(p \oplus q) \equiv -(p \equiv q)$

Au fost utilizate în ordine relațiile de subalternare (implicație), contrarietate și subcontrarietate și, în sfârșit, contradicție.
Să se construiască o interpretare corespunzătoare pentru $A = p + q$ și $I = p \vee q$. La fel, pentru $A = p \downarrow q$ și $I = -p$ și pentru $A = p$ și $I = p \subset q$.

Cap. 3. METODE DE DECIZIE ÎN LOGICA PROPOZIŢIILOR

1. Realizabilitate şi validitate

Raţionamentele făcute de oameni în limbile naturale sau în limbajele specializate ale ştiinţelor pot fi transpuse în limbaje logice (logica propoziţiilor, calculul cu predicate, în logicile intuiţioniste, în logicile modale). Pentru orice raţionament transpus într-un limbaj logic există, în principiu, tehnici şi metode de a-i verifica validitatea, de a ne asigura că din premise adevărate au fost degajate numai concluzii adevărate. Trebuie, însă, respectate două condiţii: raţionamentul să fie formalizat corect şi verificarea validităţii să se facă prin metode adecvate. Logica predicatelor este semidecidabilă. Orice raţionament valid poate fi verificat ca valid printr-un calcul finit, dar nu orice formulă invalidă poate fi atestată ca invalidă printr-un calcul finit.

Este esenţial să observăm că raţionamentele desfăşurate în conformitate cu formule logic valide conservă adevărul de la premise la concluzii şi că, pe această cale, noi accedem la concluzii mereu mai îndepărtate de baza de cunoştinţe iniţială.

În acest paragraf ne vom ocupa de metodele ce permit verificarea validităţii raţionamentelor formalizate cu ajutorul limbajului logicii propoziţiilor.

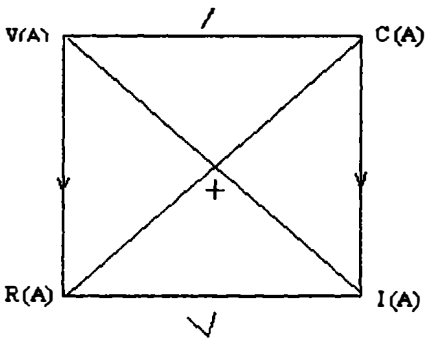
A decide asupra unei formule înseamnă a determina, pe o cale oarecare, dacă formula în cauză este: *realizabilă* sau adevărată *cel puţin* într-o interpretare; *validă* sau adevărată în *orice* interpretare; *infirmabilă* sau falsă cel puţin într-o interpretare, *inconsistentă(contradicţie)* sau falsă în orice interpretare. În logica propoziţiilor conceptul de interpretare se suprapune, grosier vorbind, cu cel de atribuire de valori de adevăr variabilelor propoziţionale.

Este util să arătăm că între cele patru calificări ce pot fi acordate unei formule testate (de regulă, descriind un raţionament sau un argument formulat într-un context dat) există următoarele raporturi:

1. Orice formulă validă este realizabilă;
2. Orice formulă irealizabilă este infirmabilă;
3. O formulă infirmabilă este invalidă;

4. O formulă realizabilă nu este inconsistentă;
5. Nici o formulă nu poate fi în acelaşi timp validă şi inconsistentă;
6. Existe formule care sunt, în acelaşi timp, realizabile şi infirmabile.

Notând validitatea prin **V**, realizabilitatea prin **ℛ**, infirmabilitatea prin **I** şi inconsistenţa sau contradicţia prin **C**, propoziţiile de mai sus pot fi ilustrate prin următorul pătrat logic:



Am folosit noţiunea de pătrat logic în accepţia sa abstractă, generalizată, după cum a fost ea definită de noi în paragraful anterior.

$$\varphi(p, q, r, s) = (p/q) \wedge (s \vee r) \wedge (p \supset s) \wedge (q \supset r) \wedge (p + r) \wedge (q + s)$$

În interpretarea dată mai sus, am postulat: p = **V**(A) , s = **ℛ**(A), q = **C**(A) , r = **I**(A), iar **V**(A) se va citi: „Formula A este validă" şi, în mod corespunzător, toate celelalte.

2. Metode semantice

2.1. **Tipologia metodelor de decizie.** De regulă, metodele de decizie folosesc simultan criterii sintactice, semantice şi algebrice. Putem, totuşi, să caracterizăm metodele de decizie, după predominanţa procedeelor utilizate, în metode: semantice, algebrice şi sintactic deductive.

Metode semantice

1. *metoda matriceală*
2. *metoda lui Quine*
3. *metoda tablourilor semantice*
4. *metoda arborilor de decizie*

Metode algebrice

1. *metoda formulelor normale(f.n.) sau canonice*

a *f.n. conjunctive*
b *f.n. disjunctive*
c *f.n. conjunctive perfecte*
d *f.n. disjunctive perfecte*

Metode sintactic-deductive

1. *metoda axiomatică*
2. *calculul natural*
3. *calculul secvenţial*
4. *metoda rezoluţiei*

Particularitatea metodelor semantice este utilizarea sistematică a atribuirilor de valori de adevăr. Sunt posibile două strategii: 1) să atribuim mai întâi valori de adevăr propoziţiilor elementare şi, apoi, din aproape în aproape, să determinăm valoarea de adevăr a agregatelor propoziţionale, utilizând pentru aceasta definiţiile date operaţiilor sau conectivelor logice; 2) să acordăm, la început ipotetic, o valoare de adevăr propoziţiei complexe asupra căreia decidem, de exemplu, falsul, şi apoi să conchidem, pas cu pas, asupra valorilor de adevăr ce revin subansamblelor acesteia, până când ajungem la atribuiri inconsistente pentru propoziţiile elementare. Pe acest temei, vom conchide, cu certitudine asupra valorii de adevăr a formulei iniţiale.

2.2. **Metoda matriceală.** În cadrul acestei metode se va utiliza prima strategie. Vom porni de la elementar spre complex.

O *matrice* este un tablou format din m × n elemente, de regulă numere, dispuse pe m linii şi n coloane, notată, frecvent, cu $A=\|a_{ij}\|$, unde i desemnează a i-a linie de sus în jos şi j a j-a coloană de la stânga la dreapta, cu $1<= i <= m$ şi $1<= j <= n$ În logică elementele matricei vor desemna valori de adevăr.

Pentru a decide prin metoda matriceală asupra unei formule din logica propoziţiilor procedăm astfel:

Identificăm, mai întâi, lista variabilelor propoziţionale care apar în formulă şi determinăm numărul acestora;

plasăm variabilele pe prima linie, în stânga, fiecare variabilă fiind în acelaşi timp începutul unei coloane;

construim cu cele n variabile toate cele 2^n atribuiri de valori distincte, câte una pe fiecare linie (sau m^n, cu m numărul valorilor de adevăr şi n numărul argumentelor sau variabilelor);

identificăm operatorul sau conectivul principal al formulei;

descompunem formula dată iniţial în subformule, până ajungem la formule constând din negaţii, conjuncţii, disjuncţii, sau implicaţii elementare şi scriem aceste formule în prima linie;

calculăm, în coloane, potrivit tabloului funcţiilor bivalente, valorile de adevăr ale propoziţiilor alcătuite din conective logice şi apoi, ascendent, valorile de adevăr ale agregatelor propoziţionale formate din acestea;

coloana de valori de adevăr obţinută pentru operatorul principal al formulei va coincide cu coloana valorilor de adevăr ale formulei testate.

Decidem asupra formulei date după criteriul: dacă coloana de valori conţine numai 1 (numai 0), atunci formula este *validă (inconsistentă)*; dacă, conţine cel puţin un 1(un 0), este *realizabilă (infirmabilă)*.

Exemplu:

$$\alpha = ((p \supset r) \wedge (q \supset r)) \supset ((p \vee q) \supset r)$$

Lista de variabile este p, q, r şi n = 3. Numărul atribuirilor de valori distincte este $2^3 = 8$. Construim plasând variabilele în stânga, pe prima linie şi, în ordine, pe aceeaşi linie, agregatele propoziţionale în ordine ascendentă. În continuare, pe liniile 2---9 scriem atribuirile de valori distincte date variabilelor şi, corespunzător, valorile calculate pentru subformule.

1	2	3	4	5	6	7	8	9
p	q	r	$p \supset r$	$q \supset r$	$p \vee q$	$(p \supset r) \wedge (p \supset r)$	$(p \vee q) \supset r$	α
1	1	1	1	1	1	1	1	1
1	1	0	0	0	1	0	0	1
1	0	1	1	1	1	1	1	1
1	0	0	0	1	1	0	0	1
0	1	1	1	1	1	1	1	1
0	1	0	1	0	1	0	0	1
0	0	1	1	1	0	1	1	1
0	0	0	1	1	0	1	1	1

Tabloul 14 11 Decizia matricială a unei formule

Se observă că valorile coloanelor 4 şi 5 s-au obţinut prin implicaţie între coloanele 1 şi 3 şi 2 şi 3 iar valorile coloanei 7 s-au obţinut prin conjuncţia dintre valorile coloanelor 4 şi 5, ale coloanei 8 prin implicaţie între 6 şi 3 şi ale lui α tot prin implicaţie intre 7 şi 8.

Şi mai simplu putem să scriem valorile din coloanele 4 şi 5 în coloana 7, în stânga şi în dreapta lui Λ, iar valorile lui 6 în coloana 8, în stânga lui $\supset$.

Observaţii asupra deciziei matriceale. Este o formă de decizie exhaustivă. Ne dă informaţii complete asupra cazurilor de adevăr şi de fals ale funcţiei, identifică toate cazurile de adevăr şi toate cazurile de fals. Este o metoda semantică ascendentă, de la simplu la compus. Pentru cazurile când $n \geq 3$, 4 devine incomodă, laborioasă şi când numărul variabilelor creşte şi mai mult devine impracticabilă cu creionul pe hârtie.

2.3. **Metoda lui W.O.Quine.** Este o metodă semantică ascendentă cu valorizări parţiale. Ea poate fi caracterizată prin următoarele instrucţiuni şi reguli:

Se identifică variabila cu numărul cel mai mare de apariţii în formula dată;

Se rescrie de doua ori formula dată, prima dată înlocuind variabila cea mai frecventă prin 1 şi a doua oară prin 0;

În fiecare dintre cele două formule se aplică regulile:

R1 Se elimină 1 dacă acesta apare ca element al unei conjuncţii;

R2 Se elimină 0 dacă acesta apare ca element al unei disjuncţii;

R3 Dacă 0 apare într-o conjuncţie, aceasta se înlocuieşte prin 0;

R4 Dacă 1 apare ca element al unei disjuncţii, aceasta se înlocuieşte prin 1.

R5 Dacă 1 apare ca antecedent al unei implicaţii, se reţine numai consecventul.

R6 Dacă 0 apare ca antecedent al unei implicaţii se înlocuieşte implicaţia cu 1.

R7 Dacă 0 apare în consecvent se reţine doar negaţia antecedentului.

R8 Se aplică regulile dublei negaţii.

Se aplică instrucţiunea 2 la următorul atom sau variabilă elementară cu cel mai mare număr de apariţii şi apoi se trece la instrucţiunile 3 şi din nou la instrucţiunea 4, până la epuizarea atomilor.

Exemplu:

$$\psi = ((\neg p \vee q) \wedge (r \supset p)) \vee (p \supset (t \vee \neg q))$$

Aplicăm instrucţiunea 1 şi stabilim că p este variabila cea mai frecventă. Conform instrucţiunii 2, rescriem formula ψ, o dată operând substituţia 1/p şi a doua oară, 0/p. Obţinem astfel:

$$\psi_1 = ((0 \vee q) \wedge (r \supset 1)) \vee (1 \supset (t \vee \neg q))$$

$$\psi_2 = ((1 \vee q) \wedge (r \supset 0)) \vee (0 \supset (t \vee \neg q))$$

Prelucrăm ψ_1 conform instrucţiunii 3, regulile R2, R6, R5, şi obţinem:

$$\psi_{12} = (q \wedge 1) \vee (t \vee \neg q)$$

Conform lui R1, ψ_{12} devine :

$$\psi_{13} = q \vee t \vee \neg q$$

care face evidentă valoarea 1. Strict, după reguli, ar urma considerarea în ψ_{13} a lui q ca cea mai frecventă variabilă şi apoi înlocuirea lui q o dată cu 1 şi a doua oară cu 0 şi constatarea că ambele cazuri sunt 1.

Prelucrarea lui ψ_2 se va face în conformitate cu regulile R4, R7 şi D1.

$$\psi_{21} = ((1 \wedge \neg r)) \vee 1$$

$$\psi_{22} = \neg r \vee 1$$

$$\psi_{23} = 1$$

Formula ψ este o tautologie.

2.4. **Metoda tablourilor semantice.** Metoda a fost creată de matematicianul olandez Evert W. Beth în deceniul al şaptelea. Este o metodă semantică de tip *Top Down*, analitic-descendentă. Se emite mai întâi ipoteza că formula dată spre testare ar fi infirmabilă, i.e. că ar exista cel puţin un model sau o interpretare pentru care formula în cauză devine falsă. Din această ipoteză, prin reguli riguroase de analiză, se trag toate consecinţele. Dacă în toate alternativele generate de aplicarea regulilor ajungem la contradicţii, atunci formula iniţială este validă.

Metoda lui Beth este o tehnica de infirmare sau de căutare a contraexemplelor. Dacă găsim un contraexemplu, formula testată este infirmabilă sau invalidă. Dacă căutarea contraexemplelor generează în toate alternativele contradicţii, formula iniţială este validă sau lege logică. In procesul analizei clasificăm subformele formulei supuse analizei în false, pe care le trecem în partea dreapta şi in adevărate, pe care le trecem în partea stângă a tabloului. Analiza este iterativă. Dăm mai întâi 8 reguli de analiză a conectivelor logice: $\wedge, \vee, \rightarrow$ *si* $\equiv$. Pentru fiecare conectiv logic vom avea două reguli de analiză: prima în domeniul adevărului; cea de-a doua în domeniul falsului.

$\wedge$ st 1	0
1.A$\wedge$B	
2. A	
3. B	

$\wedge$ dr	1		0	
			1.A$\wedge$B	
I	II		I	II
			2. A	3. B

v st	1		0	
1A$\vee$ B				
I	II		I	II
2.A	3.B			

$\vee$dr	1	0
		1.A$\vee$ B
		2.A
		3.B

$\supset$ st	1		0	
1. A$\supset$B				
I	II		I	II
2. B				3. A

$\supset$ dr	1	0
		1. A$\supset$B
I		II
2.A		3.B

$\equiv$ st	1		0		$\equiv$dr	1		0	
1. A$\equiv$B								1. A$\equiv$B	
I	II		I	II	I	II		I	II
2. A				4. A	2.A	4.B		3.B	5. A
3. B				5. B					

Regula ($\wedge$ st): din adevărul conjuncţiei A $\wedge$ B se conchide asupra adevărului lui A şi asupra adevărului lui B . Aceasta este regula eliminării conjuncţiei.

Regula ($\wedge$ dr): din falsitatea lui A $\wedge$ B se inferă asupra falsităţii lui A (prima alternativă) sau asupra falsităţii lui B (a doua alternativă).

Într-adevăr, falsitatea unei conjuncţii având doi termeni se poate datora falsităţii primului ei termen sau falsităţii celui de-al doilea termen. Ea duce la o scindare sau la considerarea a două alternative. La fel duce la scindare sau alternative regula $\vee$st sau regula disjuncţiei adevărate. Dacă este adevărat A $\vee$ B, atunci sau este adevărat A (alternativa I) sau este adevărat B (alternativa II). O disjuncţie falsă (regula $\vee$ dreapta) nu duce la alternative: se conchide cu certitudine asupra falsităţii tuturor termenilor disjuncţiei.

Regula implicaţiei în domeniul adevărului ($\supset$st) duce şi ea la alternative. O implicaţie poate fi adevărată pe temeiul că-i este adevărat consecventul sau pe temeiul că-i este fals antecedentul. Dimpotrivă, apariţia implicaţiei in domeniul falsului (0) conduce, cu certitudine, la admiterea adevărului antecedentului şi la admiterea falsităţii consecventului.

Regula echivalenţei în domeniul adevărului ($\equiv$ st) duce şi ea la scindarea tabloului iniţial: echivalenţa poate fi adevărată pe temeiul ca ambii termeni sunt adevăraţi şi pe temeiul că ambii termeni sunt falşi. Tot la scindare duce şi ($\equiv$dr): prima dată pe temeiul că este adevărat A şi fals B; a doua oară pe temeiul că este adevărat B şi fals A.

Explicaţii: În exemplul 1 s-a emis ipoteza că formula dată spre testare este infirmabilă i.e. s-a scris formula *1* în domeniul falsului. Conform regulii ($\supset$dr), din falsitatea lui 1 rezultă 2 şi 3, respectiv adevărul antecedentului şi falsitatea consecventului lui 1. În mod analog, falsitatea lui 3 duce la 4 şi 5, iar falsitatea lui 5 duce la 6 şi 7. Din analiza lui 2, care este o implicaţie în domeniul adevărului, rezultă scindarea tabloului iniţial în alternativa I în care este adevărat consecventul lui 2 şi în alternativa II în care este fals antecedentul lui 2. Astfel s-au obţinut formulele 8 şi 9.

Exemplul 1

<table>
<tr><td colspan="3">2.$(p\wedge q)\supset r$
4.p
6.q</td><td colspan="3">1.$((p\wedge q)\supset r)\supset(p\supset(q\supset r))$
3.$p\supset(q\supset r)$
5.$q\supset r$
7.r</td></tr>
<tr><td>I 8.r</td><td colspan="2">II</td><td>I</td><td colspan="2">II 9. $p\wedge q$</td></tr>
<tr><td></td><td>IIa</td><td>IIb</td><td></td><td>IIa 10.p</td><td>IIb 11.q</td></tr>
<tr><td>8-7 #</td><td>10-4 #</td><td>11-6 #</td><td>8-7 #</td><td>10-4 #</td><td>11-6 #</td></tr>
</table>

Exemplul 2

2.(p ∧ q) ∨ r 6. p≡r				1. ((p∧q)∨r) ⊃(–(p≡r)∨q) 3. –(p≡r)∨q 4. –(p≡r) 5. q			
I 7. p 8. r		II		I		II 9.p 10.r	
II 11.p∧q 15. p 16. q	I2 12.r	II1 13.p∧q 17.p 18.q	II2 14.r	I1	I2	II1	II2
16-5 #	CE	18-5 #	14-10 #	16-5 #	CE	18-5 #	14-10 #

În primul exemplu, s-a emis ipoteza că implicaţia 1 este falsă. Rezultă, deci, în conformitate cu regula ⊃ dr, că antecedentul ei, formula 2, este adevărată iar consecventul ei, formula 3, este falsă.

Propoziţiile adevărate au fost scrise în partea stângă a tabloului, iar cele false în partea dreaptă. În mod analog, este analizată, mai de-parte, implicaţia 3. La formula 2, scrisă în domeniul adevărului, se aplică regula de analiză a implicaţiei adevărate, respectiv regula ⊃ st. Formula 2 poate fi adevărată pe temeiul că este adevărat consecventul ei sau pe temeiul că este fals antecedentul ei. Aceasta duce la scinda-rea tabloului iniţial în subtablourile I şi II. În consecinţă, consecventul formulei 4 va fi scris în subtabloul I în componenta sa stângă(vezi formula 8), iar antecedentul ei, formula 9, va fi scrisă în subtabloul II, în partea dreaptă (vezi formula 9). Din falsitatea lui 9, rezultă prin (∧ dr) scindarea tabloului II în IIa şi IIb, obţinându-se astfel formulele 10 şi 11. Întrucât toate tablourile alternative au condus la contradicţii şi tablourile s-au închis, formula testată este o tautologie.

În exemplul 2, ipoteza falsităţii lui 1 duce la 2 şi la 3. Falsitatea lui 3 duce, prin (∨ dr) la 4 şi la 5. O formulă prefixată de semnul negaţiei poate fi trecută fără negaţie în partea opusă a tabloului: de la 4 s-a trecut la 6. Aplicând la 4 regula (≡st), scindăm tabloul iniţial în subtablourile I şi II din ambele domenii. Scindarea impusă de

aplicarea regulii (∨st) la formula 2 se efectuează în ambele subtablouri, ducând la formulele 11 şi 12 şi la 13 şi 14.

Subtabloul I2 nu se închide, deşi e tablou terminal. El ne indică existenţa contraexemplului căutat: p = r = 1 şi q = 0. Într-adevăr, dacă atribuim în formula 1 din exemplul 2 lui p şi r valoarea 1 şi lui q valoarea 0, formula 1, i.e. formula testată, aceasta devine falsă :

1. $((p \wedge q) \vee r) \supset (\neg (p \equiv r) \vee q)$
2. $((1 \wedge 0) \vee 1) \supset (\neg (1 \equiv 1) \vee 0)$
3. $(0 \vee 1) \supset (\neg 1 \vee 0)$
4. $1 \supset (0 \vee 0)$
5. $1 \supset 0$
6. 0

Formula testată în exemplul 2 este infirmabilă.

Reguli de închidere a tablourilor

R1. Un tablou terminal se închide, dacă şi numai dacă, conţine o pereche de formule contradictorii.

R2. Un tablou de rang n se închide dacă s-au închis toate tablourile sale imediat subordonate sau de rang n-1. De exemplu, II se închide pentru că s-au închis II a şi II b .

R3. Tabloul iniţial se închide, dacă s-au închis toate tablourile imediat subordonate lui.

Închiderea unui tablou am marcat-o prin simbolul # (vezi exemplele 1 şi 2). Totodată, am marcat închiderea prin acelaşi tip de linie în stânga şi în dreapta tabloului. Rămânerea deschisă a unui tablou sau descoperirea unui contraexemplu am notat-o prin CE (vezi exemplul 2).

2.5. Metoda arborilor de decizie. Are în esenţă acelaşi temei teoretic ca şi metoda tablourilor semantice. Ca şi metoda tablourilor semantice se întemeiază pe un raţionament apagogic sau de reducere la absurd, este analitic–descendentă, ipotetică şi porneşte, ca şi aceasta, de la ipoteza infirmabilităţii formulei supuse testării.

Diferenţa dintre cele două metode constă în principal în modul de redactare şi în utilizarea unor arbori şi bifurcaţii descendente în locul scindării tablourilor semantice. În loc de 1 şi 0 în această redactare se scrie pur şi simplu formula pentru 1 şi negaţia ei pentru 0. Introducem regulile metodei arborilor de decizie.

Metoda arborilor de decizie este o metodă de testare a realizabilităţii formulelor în logica propoziţiilor şi în logica predicatelor. A verifica realizabilitatea unei formule înseamnă a căuta sistematic un model pentru acea formulă. Dacă reuşim să descoperim un model, atunci formula este realizabilă.

a. Câteva definiţii preliminare

D1. Numim *literal* un atom sau negaţia acestuia. În logica propoziţiilor, literali vor fi variabilele propoziţionale şi negaţiile acestora. În logica predicatelor, literali vor fi atomii predicativi şi negaţiile acestora.

Exemple de literali în logica propoziţiilor: p, q, – r.

Exemple de literali în logica predicatelor: p(x), – q(a), r(x,y), – s(z,u), r(a,b). Pentru apropierea notaţie din logica predicatelor de cea din programarea logică, vom nota, în continuare, simbolurile predicative prin litere mici, ca mai sus, p(x), r(x,y) etc.

D2. Numim *pereche opusă de literali* doi atomi identici, diferind doar prin semnul negaţiei: {p, – p}, {p(a), – p(a)}, {r(a,b), – r(a,b)}.

D3. Un arbore este un graf orientat, având o rădăcină şi frunze.

Caracteristica de bază a unui arbore este absenţa ciclurilor.

D4. Numim *arbore etichetat* un arbore ale cărui noduri sunt marcate de anumite simboluri.

În continuare, vom lucra cu arbori ce vor fi etichetaţi printr-o formulă unică sau prin mulţimi structurate de formule. Aceasta va conduce la două variante ale metodei arborilor de decizie.

D5. Un drum într-un arbore este o secvenţă de arce, respectiv o secvenţă de mulţimi de formule ce etichetează nodurile arcelor, astfel că nodul terminal al arcului precedent coincide cu nodul iniţial al arcului succedent (au aceeaşi etichetă).

D6. Un *drum complet* leagă rădăcina de o frunză. Frunzele sunt etichetele prin seturi de literali.

D7. Numim tulpină a unui arbore o secvenţă de noduri şi respectiv o secvenţă de mulţimi de formule ce etichetează aceste noduri generate din rădăcina arborelui prin aplicare exclusivă a unor α - reguli.

D8. Numim ramificaţii ale unui arbore drumurile alternative generate din tulpina arborelui prin aplicarea unor β - reguli (sau reguli de tip $\vee$).

D9. Numim *nod de scindare* un nod *n* la care s-a aplicat o β - regulă. Din el vor pleca întotdeauna două arce ce conduc la nodurile succesoare n' şi n".

D10. Numim *nivel* sau înălţime a unui arbore o mulţime ordonată de mulţimi de formule ale căror componente au fost obţinute prin aplicarea aceluiaşi număr de operaţii deductibile, respectiv prin utilizarea de acelaşi număr de ori a unor α- sau β- reguli.

Prin definiţie, rădăcina are nivelul 0.

Nivelul sau *înălţimea unui arbore* este dat de nivelul sau înălţimea frunzei din cel mai lung drum. Acesta poate fi calculat astfel:

h(r)=0 (rădăcina are rangul 0).

succesor– imed(l', l) ⊃ (h(l)=m ⊃ h(l')=m+1)

(Dacă l' este succesorul imediat al nodului l şi dacă rangul lui l este m, atunci rangul lui l' va fi m+1)

b. Regulile metodei arborilor de decizie pentru logica propoziţiilor

Metoda arborilor de decizie este o metodă analitică, apagogică şi reductivă.

Este analitică pentru că pleacă de la o formulă complexă şi o descompune printr-un număr de paşi în atomi ireductibili. Este apagogică sau prin reducere la absurd pentru că emite ipoteza falsităţii formulei testate şi se încheie prin descoperirea unor contramodele, dacă formula este infirmabilă sau prin constatarea faptului că formula nu are contramodele şi deci este validă. Este reductibilă pentru că reduce problema evaluării unei formule complexe la evaluarea unor formule atomare.

Regulile utilizate de metodă presupun cunoaşterea definiţiei conectivelor logice:

$$\neg, \wedge, \vee, \supset, \subset, \equiv.$$

Metoda ne arată cum putem reduce evaluarea unei formule iniţiale la evaluarea subformulelor componente şi, în ultimă instanţă, la evaluarea atomilor componenţi. Într-o logică bivalentă, orice atom este adevărat sau fals.

Pentru fiecare operator logic binar vom avea câte o regulă pentru apariţia pozitivă a acestuia şi câte o regulă pentru apariţia sa negată.

Regulile conjuncţiei

(I) 1. $A \wedge B$
2. A
3. B

(- $\wedge$) 1. - (A $\wedge$ B)
2. -A | 3. -B

Regulile disjuncţiei

($\vee$) 1. $A \vee B$
2.A | 3.B

(-$\vee$) 1.-(A $\vee$ B)
2. -A
3. -B

Regulile implicaţiei

($\supset$) 1. A $\supset$ B
2.-A | 3.B

(-$\supset$) 1.-(A$\supset$B)
2. A
3. -B

Regulile echivalenţei:

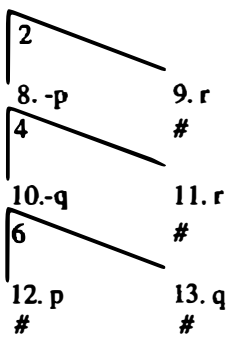
($\equiv$) 1. $A \equiv B$
2. A | 4. -A
3. B | 5. -B

(- $\equiv$) 1. -(A $\equiv$ B)
2. A | 4.–A
3. -B | 5. B

Metoda constă din enunţarea ipotezei că formula supusă examinării este infirmabilă i.e. există o interpretare care face formula falsă şi din explorarea sistematică a consecinţelor acestei ipoteze. Explorarea consecinţelor se face cu ajutorul celor opt reguli menţionate mai sus, care se aplică succesiv conectorului principal şi apoi conectivelor logice de rang mereu mai scăzut până ce se inferă asupra valorii de adevăr a propoziţiilor elementare sau variabilelor propoziţionale. Dacă un drum conţine o propoziţie (elementară) şi negaţia sa, atunci se închide. Dacă, cel puţin un drum rămâne neînchis, atunci formula testată este infirmabilă sau invalidă. Dimpotrivă, dacă se închid toate drumurile în procesul analizării descendente, formula este validă.

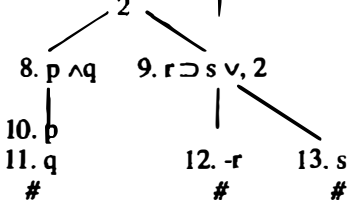
Exemplul 1:

1. -[(p ⊃ r) ⊃ ((q ⊃ r) ⊃ ((p ∨ q) ⊃ r))]
2. p ⊃ r
3. -[(q ⊃ r) ⊃ ((p ∨ q) ⊃ r))] | -⊃,1
4. q ⊃ r
5. -[(p ∨ q) ⊃ r)] | -⊃,3
6. p ∨ q
7. -r | -⊃,5



Exemplul 2:

1. -{[(p ∧q) ∨ (r ⊃ s)] ⊃ (-s ⊃(-r ∨ q))}
2. (p ∧q) ∨ (r ⊃ s)
3. -(-s ⊃ (-r ∨ q)) -⊃, 1 |
4. -s
5. -(-r ∨ q) | - ⊃, 3
6. r
7. -q | -⊃, 5
2



Regulile sunt împărţite de logicianul Ben Ari [1] în două clase: reguli ce nu duc la scindarea unui drum sau *reguli de tip conjunctiv* sau α - *reguli* şi *reguli de tip disjunctiv* ce duc la scindarea unui drum sau β - *reguli.*

Antecedent sau formulă iniţială	Consecvenţi sau formulele rezultate	
α	α_1	α_2
$--A$	A	
$(A_1 \wedge A_2)$	A_1	A_2
$-(A_1 \vee A_2)$	$-A_1$	$-A_2$
$-(A_1 \supset A_2)$	A_1	$-A_2$
$-(A_2 \supset A_1)$	$-A_1$	A_2
$A_1 \equiv A_2$	$A_1 \supset A_2$	$A_2 \supset A1$

Fig. 1 Reguli de tip conjunctiv sau α - reguli

Antecedent sau formulă iniţială	Consecvenţi sau formulele rezultate	
β	β_1	β_2
$B_1 \vee B_2$	B_1	B_2
$-(B_1 \wedge B_2)$	$-B_1$	$-B_2$
$B_1 \supset B_2$	$-B_1$	B_2
$B_1 \subset B_2$	B_1	$-B_2$
$-(B_1 \equiv B_2)$	$-(B_1 \supset B_2)$	$-(B_2 \supset B_1)$

Fig. 2 - Reguli de tip disjunctiv sau β - reguli

În lucrarea noastră ***Logica predicatelor*** (1992) am prezentat regulile sub forma unor scheme de inferenţă:

$$\frac{--A}{A} \qquad \frac{A \wedge B}{\begin{array}{c}A\\B\end{array}} \qquad \frac{-(A \wedge B)}{-A \mid -B} \qquad \frac{A \vee B}{A \mid B} \qquad \frac{-(A \vee B)}{\begin{array}{c}-A\\-B\end{array}} \qquad \frac{A \supset B}{-A \mid B} \qquad \frac{-(A \supset B)}{\begin{array}{c}A\\-B\end{array}}$$

$$\frac{A \subset B}{A \mid -B} \qquad \frac{-(A \subset B)}{\begin{array}{c}B\\-A\end{array}} \qquad \frac{A \equiv B}{A \mid -A \quad B \mid -B} \qquad \frac{-(A \equiv B)}{-(A \supset B) \mid -(B \supset A)}$$

Regulile sunt scheme de inferenţă ce conservă veridicitatea şi permit reducerea formulei iniţiale la formule atomare.

Cititorul care cunoaşte definiţiile operatorilor logici va putea lesne lucra cu aceste reguli.

Regulile metodei arborilor de decizie pot fi folosite în diferite scopuri:

pentru a verifica *validitatea* unei formule iniţiale. În acest caz, se porneşte de la negaţia formulei pe care vrem să o testăm şi căutăm sistematic contramodele. Dacă nu găsim astfel de contramodele formula este validă;

pentru a verifica *realizabilitatea* unei formule. În acest caz, supunem analizei prin reguli chiar formula a cărei realizabilitate vrem să o testăm, nu negaţia ei. Analiza se încheie prin aflarea modelelor sau interpretărilor ce fac formula iniţială adevărată. La fel putem verifica realizabilitatea sau consistenţa mutuală a unei mulţimi de formule, descriind o bază de cunoştinţe a unui sistem expert sau de inteligenţă artificială. Vom numi acest arbore – *arbore de analiză*;

pentru *a verifica echivalenţa a două formule*. În acest caz se verifică realizabilitatea ambelor formule şi dacă se ajunge la aceeaşi mulţime de modele, cele două formule sunt echivalente;

pentru *a verifica infirmabilitatea unei formule*. În acest scop supunem analizei negaţia formulei şi obţinem frunzele ei etichetate prin literali. Frunzele ce conţin o pereche opusă de literali şi sunt marcate prin semne de drum închis # ne permit să identificăm *contramodelele* sau cazurile de infirmare;

pentru a verifica în câte feluri putem *argumenta* sau *întemeia* o anumită teză logică sau pentru a afla ipotezele primare din care poate fi demonstrată teza în cauză. Pentru aceasta se analizează prin α şi β - reguli teza de argumentat şi pe fiecare drum deschis identificăm un set suficient de ipoteze iniţiale, din care poate fi derivată logic teza şi principalele argumente sau noduri ale demonstraţiei.

c. Construirea unui arbore de decizie în logica propoziţiilor. Etichetare prin mulţimi de formule

Fie F o formulă bine formată în logica propoziţiilor pe care dorim să o testăm prin metoda arborilor de decizie. Pentru aceasta procedăm astfel:

$1^{\#}$. Scriem -F în rădăcina arborelui:

Obs. 1. -F este o mulţime singleton de formule care declanşează procesul de analiză.

Obs. 2. Scrierea lui -F echivalează cu admiterea ipotezei că F este o formulă infirmabilă, adică o formulă ce poate fi falsă cel puţin într-o interpretare.

$2^{\#}$. Identificăm conectivul logic principal din F şi aplicăm regula negaţiei acestuia.

$3^{\#}$ Aplicăm, în continuare, regulile conectivelor ce apar în subformule, asociind fiecărei frunze sau nod ℓ o mulţime de subformule $U(\ell)$.

$4^{\#}$ .Condiţii de terminare: dacă $U(\ell)$ este o mulţime de literali, verificăm dacă $U(\ell)$ conţine o pereche opusă de literali; dacă conţine o astfel de pereche, atunci închidem drumul prin semnul #.

– dacă nu conţine o astfel de pereche, marcăm drumul prin O (drum deschis).

$5^{\#}$ Dacă $U(\ell)$ nu este o mulţime de literali, ci o mulţime de subformule ce conţine conective logice sau negaţii ale unor conective logice, atunci aplicăm la $U(\ell)$ α - reguli sau β - reguli. Vom da prioritate α - regulilor.

d. Regula α - formulelor

Dacă $U(\ell)$ nu este un set de literali şi conţine un α-conector, atunci creăm un nod ℓ' descendent din ℓ şi îl etichetăm prin

$$U(\ell')=U(\ell)-\{\alpha\}\cup\{\alpha_1,\alpha_2\}$$

(Dacă $A= -A_1$, nu va exista un α_2).

Obs. 3. Eticheta nodului succedent ℓ' în cazul unei α-formule va fi identică cu eticheta nodului anterior, din care se omite α-formula şi la care se adaugă componentele acesteia α_1 şi α_2.

Exemplu:

$$U(\ell) = -(A_1 \vee A_2), B, C$$
$$U(\ell') = -A_1, -A_2, B, C$$

e. Regula β - formulelor

Dacă formula aleasă din $U(\ell)$ este o β - formulă, atunci creăm două noi noduri descendente din ℓ, ℓ' şi ℓ'' având etichetele:

$$U(\ell')=U(\ell)-\{\beta\}\cup\{\beta_1\}$$
$$U(\ell'')=U(\ell)-\{\beta\}\cup\{\beta_2\}$$

Reamintim că β stă pentru antecedentul unei β reguli, de exemplu pentru $B_1 \vee B_2$, iar β_1 şi β_2 sunt argumentele operatorului disjunctiv din antecedentul β- regulii, respectiv B_1 sau B_2.

Exemplu:

$U(\ell)= \{B_1 \supset B_2, D, E\}$
$U(\ell')= \{-B_1, D, E\}$
$U(\ell'')=\{B_2, D, E\}$

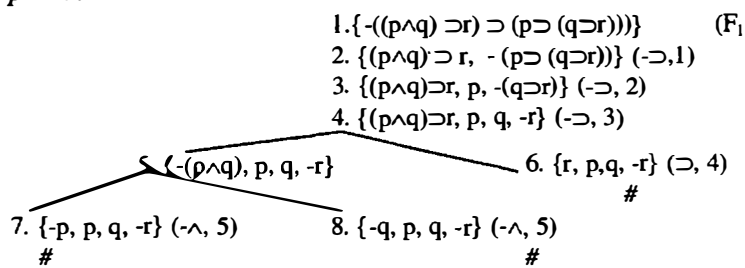
$$\frac{\{B_1 \supset B_2, D, E\}}{\{-B_1, D, E\} \mid \{B_2, D, E\}}$$

În dreapta am exprimat β - regula ca schemă de inferenţă cu două ramificaţii.

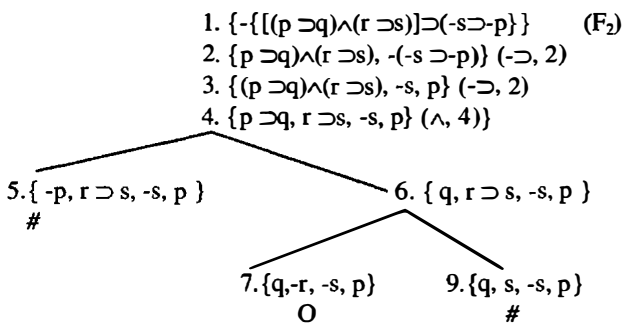
6# Construcţia arborelui se încheie când toate frunzele sunt etichetate exclusiv prin literali şi au fost marcate prin # sau O.

Un arbore este *închis* dacă toate drumurile se termină prin frunze marcate prin # şi este *deschis* dacă cel puţin un drum se termină printr-o frunză marcată prin O.

Exemplul 3:



Exemplul 4:



Obs.1 Fiecare nod este etichetat printr-o mulţime de formule. Acestea se consideră legate între ele prin ∧.

Obs.2 Formula Fl a cărei validitate este testată în exemplul 1 este legea exportaţiei: ((p∧q)⊃r) ⊃ (p⊃(q⊃r)). În rădăcina arborelui (vezi 1) am plasat pe -F, negaţia formulei testate, care este o mulţime singleton de formule. La punctele 2, 3 şi 4 am eliminat negaţia unor implicaţii, punând în locul acesteia antecedentul şi consecventul negat. -(A⊃B) este o α - regulă.

Obs.3. La punctul 4 am eliminat implicaţia $(p \wedge q) \supset r$ printr-o β- regulă, punând în locul ei subformulele $-(p \wedge q)$, r în paşii imediat următori (vezi 5 şi 6).

Obs. 4. La punctul 5 s-a aplicat o β - regulă pentru negaţia unei conjuncţii ($-(p \wedge q)$), care a generat nodurile terminale (sau frunzele) 7 şi 8, ce sunt mulţimi ce conţin perechi opuse de literali. Acestea au fost marcate prin semnul terminal de închidere a unui drum *#*. .

Obs. 5. Formula F1, testată în exemplul 3, este validă. Toate drumurile ei sunt închise: 1, 2, 3, 4, 5, 7 este, de exemplu, un astfel de drum închis.

Obs. 6. Formula F2, testată în exemplul 4, este infirmabilă. Aceasta se vede în faptul că drumul 1, 2, 3, 4, 6, 7 este un drum deschis, marcat de semnul O. Dacă $v(r)=v(s)=0$ şi $v(p)=v(q)=1$, atunci formula F2 este falsă. Verificăm aceasta prin substituirea lui r şi s prin 0 (fals) şi a lui p şi q prin 1 (adevărat).

1. $-\{(1 \supset 1) \wedge (0 \supset 0) \supset (-0 \supset -1)\}$ (F2)
2. $-\{ \quad 1 \quad \wedge \quad 1 \supset (1 \supset 0)\}$
3. $\{ \qquad 1 \qquad \supset \qquad 0 \ \}$
4. $\qquad\qquad \{0\}$

Obs. 7. Metoda arborilor de decizie este o metodă de căutare sistematică a contramodelelor. Dacă negaţia formulei este realizabilă, atunci formula în cauză este infirmabilă (are un contraexemplu).

f. Arbori completi

Un arbore poate fi în construcţie sau complet construit. Un arbore este *complet* dacă toate frunzele sale conţin numai literali.

Teorema 1. *Construcţia unui arbore de decizie pentru o formulă din logica propoziţiilor se termină după un număr finit de aplicare a regulilor de analiză.*

Demonstraţie: Fie A un arbore de decizie pentru formula F, în orice stadiu al construcţiei arborelui. Pentru orice nod în construcţie sau frunză temporară $f \in A$, fie b(f) numărul conectivelor binare din formulele E(f) ce etichetează frunza f şi fie n(f) numărul negaţiilor din E(f).

Definim numărul argumentelor (subformulelor) frunzei f:

$$Arg(f)=2b(f)+n(f)$$

Cum aplicarea unei reguli generează una sau două submulţimi de formule, f ' pentru regulile de tip α şi f ' şi f '' pentru regulile de tip β,

putem arăta că Arg(f ')<Arg(f) şi Arg(f')<Arg(f). Cu fiecare aplicare a unei reguli, gradul de complexitate a formulelor rămase scade.

Să admitem că aplicăm la f o regulă de tip α, de exemplu, regula pentru negaţia unei implicaţii -(A1⊃A2). Atunci:

Arg(f ')=k+0+2 < k+2+1=Arg(f), unde k este suma numerelor de conective care apar în A1 şi A2. Cum ∞ > Arg(f), 3>0, nici o ramificaţie a lui f nu poate fi extinsă infinit.

Dacă în f apare ≡, acesta poate fi redus la (A1⊃A2)∧(A2⊃A1), acestea putând fi, în continuare, reduse prin regula eliminării conjuncţiei (o α- regulă) şi prin regulile eliminării implicaţiilor (β- reguli).

Notă: Este uşor de observat că:

Arg(-(A1⊃A2)) = k + 2b(-(A1⊃A2)) + n(-(A1⊃A2)) = k+2+1= k+3.

Arg(A1, -A2) = k+0+1 = k+1, Arg(f) = k+3. Arg(f ') = k+1.

Deci, Arg(f ') < Arg(f).

De aici rezultă că orice analiză a unui set iniţial de formule din logica propoziţiilor, efectuată cu ajutorul regulilor de reducţie de tip α sau β se termină după un număr finit de paşi, producând ca rezultat final un număr finit de literali pe orice frunză sau nod teminal în arbore.

Obs. 8. Metoda arborilor de decizie este o *metodă de reducere* a unei formule sau set de formule la un set de literali sau atomi cu sau fără negaţie i.e. variabile propoziţionale evaluate din punctul de vedere al veridicităţii. Ea identifică temeiurile ultime ale veridicităţii formulei iniţiale; ea specifică în ce condiţii formula iniţială poate fi adevărată. Ea descoperă, aşadar, modelele sau circumstanţele în care formula este adevărată.

Obs. 9. Metoda arborilor de decizie poate deveni relevantă din punctul de vedere al teoriei argumentării. A argumenta o teză înseamnă a specifica temeiurile şi condiţiile care o fac adevărată. A argumenta înseamnă a identifica temeiurile (legile şi faptele) care fac o teză adevărată.

Argumentarea se face prin referire la un *sistem teoretic* sau prin referire la *un sistem normativ* şi la o bază factuală din care decurge o altă propoziţie factuală în disputa a doi sau mai mulţi interlocutori.

g. Arborii de decizie. Realizabilitatea, infirmabilitatea şi validitatea

Metoda arborilor de decizie permite evidenţierea unor proprietăţi sintactice şi semantice ale formulelor din logica propoziţiilor, cum ar fi realizabilitatea, infirmabilitatea, validitatea şi irealizabilitatea sau inconsistenţa mutuală a unui set de formule. Dăm, mai jos, câteva teoreme despre aceste proprietăţi.

Teorema 2. *O formulă F este realizabilă, dacă şi numai dacă, arborele ei de decizie este deschis.*

Fie F o formulă şi A arborele ei de decizie deschis. Trebuie, deci, să demonstrăm mai întâi teza:

a) Dacă arborele de decizie al formulei este deschis, atunci formula F este realizabilă

Admitem, deci, ipoteza:

Arborele de decizie A al formulei F are cel puţin un drum deschis şi o frunză ca nod terminal în acel drum, etichetată printr-un set de literali ce nu conţine o pereche opusă de literali. Eticheta nodului terminal al drumului deschis din A este de forma E(f)={p, -q, r} unde p, q, r sunt variabile propoziţionale sau atomi ce apar în F.

Dacă vom asocia etichetei valorizarea v(p)=1, v(q)=0 şi v(r)=1, vom găsi o interpretare pentru care formula F este adevărată, iar o formulă care are o interpretare ce o face adevărată este realizabilă.

Să demonstrăm teza conversă.

b). Dacă F este o formulă realizabilă, atunci arborele ei de decizie este deschis.

Admitem deci ipoteza: F este o formulă realizabilă.

Plasăm formula F în rădăcina arborelui şi aplicăm asupra ei α- şi β- regulile de analiză.

Formula F fiind realizabilă, va avea cel puţin o interpretare în virtutea căreia devine adevărată. În logica propoziţiilor, acea interpretare va avea forma {v(p)=1, v(q)=1, v(r)=0} sau alta similară. Unei astfel de interpretări putem lesne să-i asociem o secvenţă de literali {p, q, -r} care etichetează o frunză a unui drum în A.

O etichetă ce nu conţine o pereche opusă de literali marchează un drum deschis. Un arbore ce are un drum deschis este, în mod necesar, un arbore deschis. Din punct de vedere tehnic demonstraţia de mai sus poate fi descrisă printr-o inducţie matematică pe etapele construirii arborelui de decizie asociat formulei în cauză.

Prima parte (vezi mai sus punctul a)) presupune parcurgerea descendentă sau analitică a arborelui de decizie. Dimpotrivă, partea a

doua presupune parcurgerea ascendentă, de la frunze sau model la formula iniţială. Aceasta are un caracter sintetic-constructiv.

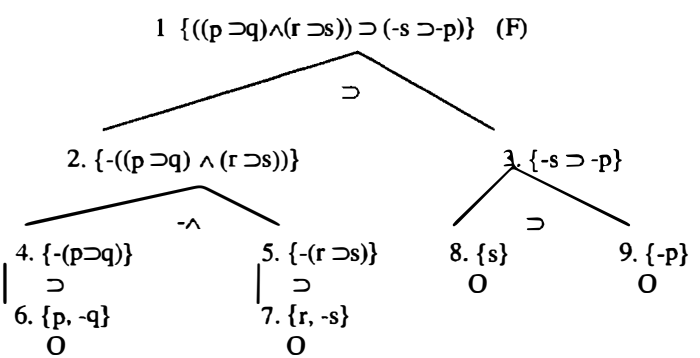
Este posibilă o parcurgere bidirecţională descendentă sau analitică şi ascendentă sau sintetică, deoarece toate α- şi β- regulile din metoda arborelui de decizie pot fi descrise ca echivalenţe:

α - reguli:
- $v(- -A) \equiv v(A)$
- $v(A1 \wedge A2)=1 \equiv v(A1)=1$ şi $v(A2)=1$
- $v(-(A1 \vee A2)=1 \equiv v(A1)=0$ şi $v(A2)=0$ sau
- $v(-A1)=1$ şi $v(-A2)=1$
- $v(-(A1 \supset A2)=1 \equiv v(A1)=1$ şi $v(A2)=0$ i.e. $(v(-A2)=1)$
- $v(A1 \equiv A2)=1 \equiv v(A1 \supset A2)=1$ şi $v(A2 \supset A1)=1$

β - reguli:
- $v(B1 \vee B2)=1 \equiv v(B1)=1$ sau $v(B2)=1$
- $v(-(B1 \wedge B2))=1 \equiv v(B1)=0$ sau $v(B2)=0$, i.e. $(v(-B1)=1$ sau $v(-B2)=1)$
- $v(B1 \supset B2)=1 \equiv v(B1)=0$ sau $v(B2)=1$, $(v(-B1)=1)$
- $v(-(B1 \equiv B2))=1 \equiv v(B1 \supset B2)=0$ sau $v(B2 \supset B1)=0$
- $v(-(B1 \supset B2))=1$ sau $v(-(B2 \supset B1))=1$

Folosind demersul descendent descris mai sus, vom identifica, mai întâi, mulţimea modelelor unei formule din logica propoziţiilor.

Vom considera pentru aceasta tot formula din exemplul 2. Pornim însă de la ipoteza adevărului ei.



Fiecare dintre frunze descrie o interpretare parţială sau o clasă de modele în care formula iniţială F devine adevărată. Modelele formulei F sunt: {p, -q}, {r, -s}, {s}, {-p}, respectiv {v(p) = 1, v(q) = 0}, {v(r) = 1, v(s) = 0} etc.

Interpretările parţiale pot fi lesne expandate în interpretări complete, prin adăugarea în fiecare a atomilor care lipsesc în conformitate cu legile algebrei booleene.

Metoda arborilor de decizie poate fi folosită şi pentru a afla forma normală disjunctivă a formulei. Pentru aceasta, ca mai sus, se aplică α şi β - regulile direct la formula iniţială F, iar formulele din etichetele frunzelor se leagă în interior prin „şi", iar între ele prin „sau"

Forma normală disjunctivă a formulei F de mai sus este: $p\neg q \vee r\neg s \vee s \vee \neg p$.

Putem, totodată, calcula uşor din forma normală disjunctivă şi forma normală conjunctivă:– $q \vee r \vee s \vee \neg p$. De aici putem conchide că F va fi infirmată doar prin negaţia formei normale conjunctive, respectiv prin $pq\neg r\neg s$ ceea ce concordă cu testul efectuat în exemplul 2, unde negaţia formulei F a avut ca model pe $\{-r, -s, p, q\}$.

Parcurgerea ascendentă a drumurilor din arborele asociat formulei din exemplul 2 poate fi ilustrată prin tot atâtea raţionamente constructive descrise de regulile ce intervin în drumul considerat.

De exemplu, drumul ascendent 6, 4, 2, 1 din arborele formulei F de mai sus descrie raţionamentul următor:

6 $p, -q$

$\dfrac{}{-(p \supset q)}\ (-\supset)$

$\dfrac{}{-[(p \supset q)\wedge(r \supset s)]}\ Y$

2 $-[(p \supset q)\wedge(r \supset s)]$

$\dfrac{}{-[(p \supset q)\wedge(r \supset s)]\vee(-s \supset -p)}\ \vee$

2' $-[(p \supset q)\wedge(r \supset s)]\vee(-s \supset -p)$

$\dfrac{}{[(p \supset q)\wedge(r \supset s)] \supset (-s \supset -p)}\ \supset$

1 $[(p \supset q)\wedge(r \supset s)] \supset (-s \supset -p)$

Regulile aplicate sunt banale:

$\supset$ Dacă antecedentul este adevărat şi consecventul este fals, implicaţia este falsă (6 - 4);

$\wedge$ Dacă un termen al unei conjuncţii este fals, conjuncţia este falsă;

$\vee$ Dacă un termen al unei disjuncţii este adevărat, disjuncţia este adevărată;

$\supset$ Dacă este fals antecedentul sau este adevărat consecventul, implicaţia este adevărată.

Exercițiul de mai sus este concludent pentru cercetarea corelației dintre metoda arborilor de decizie (sau a tablourilor semantice) și calculul natural sau secvențial.

Sugerăm cititorului să construiască arborele de decizie pentru formula:

$$[(p \supset q) \wedge (-r \supset -q)] \wedge p)] \supset r$$

și să descrie riguros raționamentele constructive descrise de parcurgerea ascendentă a drumurilor deschise din arbore.

Teorema 3. *O formulă F este infirmabilă, dacă și numai dacă negația ei, -F, are cel puțin un drum închis, respectiv o frunză etichetată printr-o mulțime contradictorie de literali.*

Demonstrația se face prin inducție pe graful de analiză și/sau al negației formulei testate.

O formulă care are toate drumurile închise va fi infirmabilă în fiecare nod terminal și deci irealizabilă. Negația ei va fi o formulă validă.

Pe baza legii $(A \equiv B) \supset (-A \equiv -B)$ putem reformula teorema 2 de mai sus ca:

Teorema T2*: *O formulă F este irealizabilă, dacă și numai dacă arborele ei de decizie este închis.*

Cum negația unei formule irealizabile este o formulă validă, putem scrie: dacă arborele A al formulei F este închis, atunci formula F este irealizabilă și negația ei, -F, este o formulă validă. Altfel spus, din teorema T2* putem degaja, deopotrivă, teorema de validitate și teorema de completitudine a metodei arborilor de decizie, în raport cu formulele logicii propozițiilor.

T2.1*(*validitate*): Dacă arborele negației unei formule F este închis, atunci formula F este *validă*

T2.2*: Dacă formula F este validă, atunci arborele negației sale este închis (*completitudine*).

Propoziția T2.1* spune că metoda arborilor de decizie este un test de identificare a formulelor valide.

Propoziția T2.2* spune că pentru orice formulă validă putem construi negația ei și pentru acesta un arbore de decizie închis.

Cu alte cuvinte, *nu există nici o formulă validă pentru care să nu putem construi un arbore de decizie închis.*

Teorema T2.1.* (*soundness theorem*): *Dacă negația unei formule are un arbore de decizie închis, atunci formula inițială, din care s-a obținut negația, este validă.*

Fie F o formulă, astfel că -F are un arbore de decizie închis. Aceasta înseamnă că -F este irealizabil.

Un arbore de decizie este închis, dacă toate „frunzele" sale sunt închise. „Frunzele" sunt nodurile terminale ale drumurilor într-un arbore de decizie.

Plasăm formula -F în rădăcina arborelui de decizie. -F va fi o mulţime singleton de formule ce eticheteză rădăcina arborelui. Toate nodurile sunt generate din nodul iniţial (rădăcina), prin aplicarea unor reguli de tip α sau de tip β.

Dacă vom da prioritate regulilor de tip α, atunci vom aplica la formulele ce desemnează eticheta unui nod o regulă de tip β, numai dacă la acesta nu se mai poate aplica nici o regulă de tip α. Această strategie va crea „tulpina" arborelui de decizie *şi va amâna, pe cât posibil*, scindarea tulpinii în ramuri şi a ramurilor în subramuri.

Admitem că rădăcina unui arbore are înălţimea 0. Fie r = rădăcina arborelui şi h un simbol pentru înălţime. Dacă h(r) = 0 şi arborele este complet, atunci r este şi frunză sau nod terminal. Dacă r este un nod terminal închis, atunci eticheta lui r conţine o pereche opusă de literali de forma {p, -p} şi, deci, pentru orice valorizare v, v(p) = 1 şi v(-p) = 1, ceea ce este, evident, o contradicţie. Deci, E(r) este irealizabilă.

Să admitem acum că avem un nod n cu înălţimea h(n) = m şi mulţimea de formule E(n).

Cazul 1. Dacă la nodul n s-a aplicat o regulă de tip α, atunci mulţimea E(n) va conţine un operator de tip α (de exemplu, $\wedge$ sau negaţia implicaţiei $-(A1 \supset A2)$). $E(n) = \{A1 \wedge A2, C\}$, h(n) = m, iar nodul succesor lui n, n' va fi etichetat prin mulţimea $E(n') = \{A1, A2, C\}$, h(n') = m+1.

Admitem, prin ipoteză, că nodul n' este închis. Atunci E(n') va conţine o pereche opusă de literali de forma {A1, -A1} şi, deci, pentru orice valorizare v, va fi fals fie A1 fie -A1.

Dacă vom avea în E(n'), v(A1) = 0 sau v(A2) = 0, atunci cu certitudine vom avea, mai sus, în E(n), $v(A1 \wedge A2) = 0$ şi, deci, v(E(n))=0 (căci mulţimile de formule sunt legate prin $\wedge$) şi, deci, E(n) va fi irealizabil.

Cazul 2. Dacă la nodul n s-a aplicat o regulă de tip β, şi deci E(n) a conţinut un β - operator (pe $\vee$ sau $\rightarrow$) i.e. $B1 \vee B2$, atunci, ca urmare a acestuia au fost generate nodurile divergente n' şi n'', astfel că:

$E(n') = \{B1, C\}$ $\quad$ h(n') = m+1

$E(n'') = \{B2, C\}$ $\quad$ h(n'') = m+1

Admitem, prin ipoteză, că nodurile n' şi n'' sunt închise. Prin regula de închidere, etichetele vor conţine o pereche opusă de literali sau vor fi falşi deopotrivă B1 şi B2, respectiv vom avea v(B1) = 0 şi v(B2) = 0.

Prin regula falsităţii disjuncţiei, vom putea conchide asupra falsităţii lui B1∨ B2 din nodul precedent, v(B1∨B2) = 0 şi, în consecinţă, asupra irealizabilităţii formulei cu β - operator.

Dar E(r) = -F, h(r) = 0. Deci -F este irealizabil. Prin urmare, o formulă F a cărei negaţie -F este irealizabilă va fi, în mod necesar, validă.

Prin urmare, *pentru ca o formulă să fie validă este suficient ca negaţia ei să aibă un arbore de decizie închis.*

Vom reţine, de asemenea, că dacă toate frunzele unui nod n sunt închise, atunci eticheta acelui nod, E(n), este irealizabilă. *Irealizabilitatea se propagă de la frunze la rădăcini prin regulile negării conjuncţiei (α- reguli) sau prin regula negării disjuncţiei (β - reguli).*

Teorema 2.2* (*completitudine*): *Dacă o formulă F este validă, atunci negaţia ei, -F, va avea un arbore de decizie închis.*

Admitem că formula F este validă. Atunci negaţia ei -F va fi o formulă irealizabilă. Orice formulă validă adusă la forma normală conjunctivă va conţine în fiecare termen al conjuncţiei o pereche opusă de literali. Negaţia formei normale conjunctive a unei tautologii va duce la o disjuncţie de contradicţii, respectiv la o disjuncţie de mulţimi de literali ce conţin perechi opuse.

Plasăm această mulţime de literali incompatibili în frunzele acestui arbore.

Construim ascendent, rând pe rând, ramurile, ramificaţiile şi tulpina arborelui, utilizând în sens invers β şi α regulile. La început vom da prioritate β - regulilor de forma:

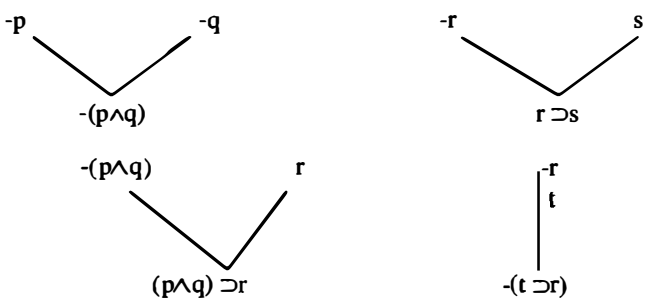


Fig. 3 Reguli ascendente

Construcţia se încheie prin obţinerea negaţiei unei formule valide.
T2.2.* se mai numeşte şi demonstraţie de completitudine.

h. Mulţimi Hintikka

Def. 5.1. Fie U o mulţime de formule şi A1, ..., An atomii care apar în formulele U.

Numim *mulţime Hintikka* o mulţime de formule care satisfac condiţiile:

1. Pentru orice atom Ai, $1 \leq i \leq n$, $Ai \in U$ sau $-Ai \in U$;

2. Dacă A este o α - formulă ce apare în U (ca $A1 \wedge A2$), atunci $A1 \in U$ şi $A2 \in U$;

3. Dacă B este o β - formulă ce apare în U (ca $B1 \vee B2$), atunci $B1 \in U$ sau $B2 \in U$;

4. Dacă A este o γ-formulă ca $\forall x A^I(x)$, atunci $A^I(a) \in U$ pentru toate constantele individuale din U;

5. Dacă A este o δ - formulă ca $\exists x A^I(x)$, atunci $A^I(a) \in U$ pentru cel puţin o constantă a din U.

Condiţiile 4 şi 5 se adaugă pentru formulele din logica predicatelor. Pentru cele din logica propoziţiilor sunt suficiente condiţiile 1, 2, 3.

Teorema 4. *Dacă M este o mulţime Hintikka de formule, atunci M este realizabilă.*

Fie A1, ..., An atomii care apar în mulţimile M de formule.

Utilizând atomii de mai sus, definim o interpretare pentru formulele din M, după cum urmează:

$v(Ai) = 1$, dacă $Ai \in M$ sau $-Ai \mathrm{I} M$

$v(Ai) = 0$, dacă $-Ai \in M$, pentru orice i, $1 \leq i \leq n$.

Revenind la definiţia unei mulţimi Hintikka (vezi D 5.1), observăm că interpretarea de mai sus satisface exigenţa 1 postulată în definiţia mulţimilor Hintikka.

Într-o mulţime Hintikka poate fi inclus un singur literal dintr-o pereche opusă de literali (vezi condiţia 1 în D 5.1). Această condiţie exclude contradicţia la nivelul literalilor.

Celelalte două condiţii stipulate pentru logica propoziţiilor în definiţia mulţimii Hintikka descriu reguli de analiză a formulelor compuse prin α - reguli sau β - reguli şi menirea lor este de a reduce formulele complexe la formule simple sau literali.

Cu ajutorul acestor reguli, orice mulţime Hintikka poate fi redusă printr-un număr finit de paşi (vezi T1) la o listă de literali fără perechi opuse. Or, valorizarea acestor liste, potrivit interpretării menţionate mai sus, conduce direct la identificarea unei interpretări I în care mulţimea de formule M este adevărată.

Formal, raţionamentul de mai sus poate fi redat prin regulile:

1. Dacă $F \in M$ este un atom, i.e. F = Ai, atunci v(F) = v(Ai) = 1;

2. Dacă $F \in M$ este negaţia unui atom, i.e. F= -Ai şi acesta nu apare în M, atunci v(F) = v(Ai) = 1;

3. Dacă F este o α - operaţie de forma $F = A1 \wedge A2$ şi v(F) = 1, atunci prin condiţia 2 (vezi D10), $A1 \in M$ şi $A2 \in M$ şi v(A1) = 1 şi v(A2) = 1;

4. Dacă F este o β - formulă i. e. $F = A1 \vee A2$ şi v(F) = 1, atunci, prin condiţia 3 (vezi D 5.1) vom avea într-o variantă $A1 \in M$, iar în cealaltă $A2 \in M$ şi, prin urmare, v(A1) = 1 sau v(A2) = 1.

Aceste reguli sunt suficiente pentru reducerea tuturor formulelor F din M la unul sau mai multe modele ale lor.

Or, un set de formule M care are cel puţin un model (i.e. o interpretare pentru care devine adevărat) este realizabil.

Un exemplu de mulţime Hintikka:

O mulţime M de formule Hintikka este o mulţime de expresii booleene i.e. de formule de logica propoziţiilor sau din logica predicatelor care nu conţine vreo pereche opusă de literali şi este închisă faţă de regulile de eliminare a conjuncţiei, disjuncţiei, cuantificatorului existenţial şi cuantificatorului universal. Aplicarea acestor reguli conduce în mod firesc la construirea unui model pentru setul iniţial de formule.

1. $\{(p \wedge q), (r \vee s), ((t \wedge v) \vee (-r \wedge w))\} = M$;
2. $\{p, q, r \vee s, ((t \wedge v) \vee (-r \wedge w))\}$ $E\wedge$, 1;
3. $\{p, q, r, (t \wedge v) \vee (-r \wedge w)\}$ $E\vee$, 2;
4. $\{p, q, s, (t \wedge v) \vee (-r \wedge w)\}$ $E\vee$, 2;
5. $\{p, q, r, t \wedge v\}$ $E\vee$,3;
6. $\{p, q, r, (-r \wedge w)\}$ $E\vee$,3;
7. $\{p, q, s, t \wedge v\}$ $E\vee$,4;
8. $\{p, q, s, -r \wedge w\}$ $E\vee$,4;
9. {p, q, r, t, v} $E\wedge$,5;
10. { p, q, r, -r, v} $E\wedge$, 6;
11. {p, q, s, t, v } $E\wedge$, 7;
12. {p, q, s, -r,w} $E\wedge$8;
13. {{p, q, r, t, v},{p, q, s, t, v}{p, q, s, -r, w}}

Formulele 9, 11, 12 exprimă modelele mulţimii M. La 13 am listat mulţimea modelelor lui M.

Pe scurt, teorema 4 afirmă că reuniunea mulţimilor ce etichetează un drum deschis este o mulţime Hintikka.

Demonstraţie:

M este reuniunea mulţimilor de formule ce etichetează nodurile unui drum deschis ce se termină în frunza f.

Cum drumul este deschis, eticheta lui f va fi o mulţime de literali ce nu conţine o pereche opusă de literali.

O mulţime Hintikka satisface în logica propoziţiilor primele trei condiţii din D 5.1. Ea nu conţine o pereche opusă de literali. Pentru orice literal Ai ori Ai∈M ori -Ai∈M. M nu le conţine nicidecum pe ambele. Această primă condiţie este satisfăcută de nodul terminal E(f).

Să presupunem că o formulă F = (A1∧A2)∈M etichetează un nod n din drum. Atunci printr-o α - regulă corespunzătoare, vom avea un nod succesor n', A1∈E(n') şi A2∈E(n'). Prin aceasta va fi satisfăcută condiţia 2 din D10.

Să presupunem că o formulă F = (B1∨B2)∈M etichetează un nod n. Atunci, printr-o β - regulă, B1∈E(n') sau B2∈E(n''), unde n' şi n'' sunt două noduri succedente imediat din n. Dar prin aceasta va fi satisfăcută condiţia 3 din definiţia D 5.1.

Modul de generare a mulţimii de formule M ce etichetează nodurile unui drum deschis satisface primele trei cerinţe din caracterizarea unei mulţimi Hintikka, valabile pentru logica propoziţiilor (vezi D 5.1) şi, în consecinţă, mulţimea M este o mulţime Hintikka.

Mai departe, pe baza teoremei 4 şi a teoremei 3, vom putea conchide că mulţimea M de formule ce etichetează un drum deschis dintr-un arbore de decizie este realizabilă.

Noţiunea de mulţime Hintikka a unui drum deschis se va extinde pentru logica predicatelor.

i. Arborii de decizie şi mulţimile Hintikka

Un exemplu: Fie F = [(p ⊃ q)∧(-r ⊃ -q) ∧ p] ∧ r o formulă din logica propoziţiilor şi A un arbore de analiză al lui F. Reamintim că un *arbore de analiză* este un arbore de decizie în care se pleacă de la formula iniţială nenegată.

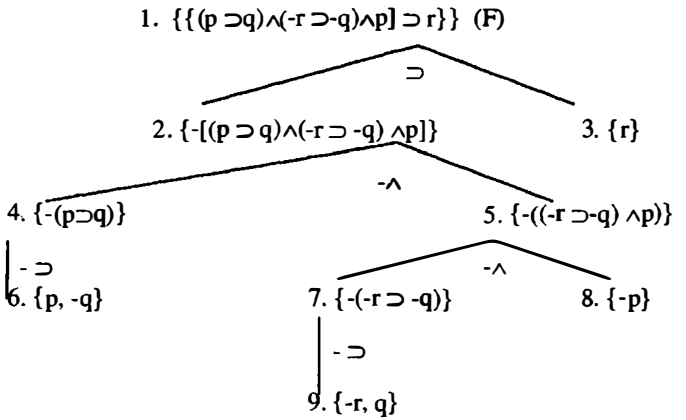


Fig. 4. Arborele de analiz• A al formulei F

Identificăm mulţimile de formule Hintikka puse în evidenţă de arborele de analiză A:

H_1={{p, -q}, {-(p⊃q)}, {-[(p⊃q)∧(-r⊃q)∧p}, {[(p⊃q)∧(-r ⊃ -q)∧p] ⊃ r}}

H_2={{-r, q}, {-(-r ⊃-q)}, {-((-r ⊃-q)Yp)}, {-[(p⊃q)Y(-r ⊃-q)Yp]},
{[(p⊃q)∧(-r⊃-q)∧p] ⊃r}}

H_3={{-p}, {-(-r ⊃-q)∧p)}, {-[(p⊃q)∧(-r ⊃q)∧p}, {[p⊃q)∧(-r ⊃-q)∧p] ⊃ r}}

H_4={{r},{[(p⊃q)∧(-r⊃-q)∧p] ⊃r}}

1. Mulţimile H_1, H_2, H_3, H_4 sunt mulţimi Hintikka. Fiecare dintre ele cuprinde etichetele unui drum deschis, enumerate de la frunză spre rădăcină.

2. Etichetele frunzelor descriu modele ale formulei în cauză, din acceptarea cărora rezultă logic adevărul formulei F. Specificăm derivarea formulei F din modelul specificat în H_2.

-r, q
——————————— -⊃
-(-r ⊃-q)
——————————— -∧
-[(-r ⊃-q)∧p]
——————————— -∧
-[(p⊃q)∧(-r ⊃-q)∧p]
——————————— ∨
-[(p⊃q)∧(-r⊃-q)∧p]]∨ r
——————————— ⊃
[(p⊃q)∧(-r⊃-q)∧p] ⊃ r

3. Formulele ce etichetează un nod sunt legate prin „şi". Mulţimile de formule succedente aceluiaşi nod sunt legate prin „sau"

4. Disjuncţia etichetelor frunzelor într-un arbore de analiză formează forma normală disjunctivă a nodului iniţial, respectiv a rădăcinii. Forma normală disjunctivă a formulei F este:

$$p\text{-}q \vee \text{-}rq \vee \text{-}p \vee r$$

Regulile de analiză utilizate mai sus, sunt împărţite în α - reguli, ce nu duc la scindări şi în β - reguli ce duc la scindări şi determină condiţiile veridicităţii formulei analizate.

Dacă formula iniţială este validă (vezi cazul din exemplul anterior), va fi validă şi disjuncţia dintre conjuncţiile atomilor ce etichetează frunzele, respectiv forma normală disjunctivă a formulei analizate.

Dacă formula iniţială este realizabilă, va fi realizabilă şi disjuncţia etichetelor frunzelor.

Dacă formula iniţială este infirmabilă, cel puţin o frunză va fi etichetată de o pereche opusă de literali.

Dacă formula iniţială este irealizabilă, atunci toate frunzele vor fi etichetate prin perechi opuse de literali.

Arborii de analiză pot fi parcurşi descendent şi ascendent.

Teoria arborilor de decizie trebuie conectată cu calculul natural şi cu cel secvenţial.

Am prezentat mai sus două variante ale metodei arborilor de decizie. O primă variantă în care un nod într-un arbore este etichetat de către o singură formulă şi o a doua variantă în care un nod într-un arbore este etichetat de o mulţime de formule. În prima variantă, fiecare subformulă dintr-o mulţime Hintikka de formule aflată pe un drum terminal într-un arbore este denumită de un număr natural distinct; în cea de a doua, nu dispunem de etichete numerice distincte pentru fiecare subformulă. Ambele variante conduc la identificarea aceloraşi contramodele, dacă există, şi la acelaşi rezultat final în actul de decizie.

Cap. 4. CALCULUL SECVENŢIAL

1. Consecinţa logică şi calculul secvenţial

Prezentăm mai jos calculul secvenţial fără regula tranzitivităţii (*cut free sequent calculus*).

Sunt presupuse ca asimilate cunoştinţele despre limbajul logicii propoziţiilor şi semantica acestui limbaj.

Noţiuni preliminare

Calculul secvenţial este o teorie formală a deducţiei şi a argumentării. În centrul preocupărilor sale se află relaţia de consecinţă logică generalizată „Din A_1 şi...şi A_n se deduce B_1 sau...sau B_n", exprimată sub forma unui secvent.

Un secvent este o pereche ordonată de mulţimi de formule (Γ, Δ) şi va fi notat prin $\Gamma \rightarrow \Delta$, unde Γ este *antecedentul,* iar Δ este *succedentul*. Este posibil ca antecedentul sau succedentul să fie nule, respectiv să avem (Φ, Δ) sau (Γ, Φ). Aceste cazuri vor fi notate prin $\rightarrow \Delta$ şi respectiv $\Gamma \rightarrow$. Convenim să notăm prin Γ_1, A mulţimea $\Gamma_1 \cup \{A\}$ şi prin A, Δ mulţimea $\{A, \Delta\}$.

Un secvent poate fi interpretat semantic prin referire la noţiunea de *realizare* într-un model de ordinul întâi.

$\Gamma \rightarrow \Delta$ este *adevărat* într-un model M, respectiv $M \models \Gamma \rightarrow \Delta$, dacă ori de câte ori sunt adevărate toate formulele A din Γ într-un model M, există cel puţin o formulă B din Δ care este şi ea adevărată.

Formulele din antecedentul secventului sunt legate prin $\wedge$, iar cele din consecvent sunt legate prin $\vee$. Dacă Γ este mulţimea $\{A_1 ....A_n\}$ şi Δ este mulţimea $\{B_1 ....B_n\}$ atunci putem exprima secventul $\Gamma \rightarrow \Delta$ prin formula :

$$A_1 \wedge ... \wedge A_n \rightarrow B_1 \vee ... \vee B_n$$

iar *interpretarea semantică a secventului* poate fi redată prin expresia:

$$M \models \Gamma \rightarrow \Delta \Leftrightarrow \forall A\,(A \in \Gamma \wedge M \models A) \rightarrow \exists B(B \in \Delta \wedge M \models B).$$

În calculul secvenţial creat de Gentzen interveneau trei tipuri de reguli: reguli iniţiale sau axiome, reguli logice şi reguli structurale. Γ şi

Δ erau interpretate ca secvențe sau șiruri de formule. În interpretarea pe care o adoptăm mai jos Γ și Δ vor fi mulțimi de formule și în consecință vom putea exclude regulile structurale, căci proprietățile acestora rezultă din teoria mulțimilor.

La Gentzen, unica formulă inițială era principiul identității: $A \rightarrow A$, „din A se deduce A" În formularea pe care o prezentăm, axiomele sau regulile inițiale vor avea forma:

$\Gamma, A \rightarrow A, \Delta$ (axiome), respectiv singura cerință pusă asupra unei axiome va fi existența unei formule ce apare, deopotrivă, în antecedentul Γ și în succedentul Δ, i.e. $\Gamma \cap \Delta \neq \Phi$.

În calculul secvențial vor fi date reguli logice pentru conectivele logice, pentru cuantificatori și pentru definițiile inductive, câte una pentru introducerea acestora în antecedent și câte una pentru introducerea acestora în succedent.

Simbolul a cărui regulă se introduce se numește *simbol principal* și formula în care apare acesta *formulă principală*.

Argumentele simbolului principal dintr-o formulă principală se numesc *subformule principale*.

O formulă ce apare într-un secvent se numește *s-formulă* (secvent-formulă). Dacă s-formula apare în antecedent se va numi *antecedent s-formula*. Similar, dacă formula apare în succedent se va numi *succedent s-formulă*.

Secvenții aflați deasupra liniei unei reguli se vor numi *premise* ale regulii, iar secventul aflat sub linia unei reguli se va numi *concluzia* regulii.

Utilizarea unei reguli logice se va numi *inferență*, dacă se face trecerea de la premise la concluzie (de sus în jos) și se va numi *reducție* sau *întemeiere*, dacă se trece de la concluzie la premise (de jos în sus).

Apariția în premise a subformulelor principale ale unei reguli se numesc formule laterale (*side s-formulas*) ale inferenței sau reducției.

O *derivare* în calculul secvențial va fi redată sub forma unui arbore rezultat din aplicarea regulilor logice.

Secventul aflat la rădăcina derivării se numește *secvent terminal*; el descrie teza de demonstrat. Secvenții aflați la capetele ramificațiilor superioare ale arborelui se numesc *frunze* sau *secvenți inițiali*.

O *demonstrație* constă din derivarea prin reguli logice a unui secvent terminal din secvenți inițiali.

O *argumentare* sau întemeiere a unei teze va consta în parcurgerea de jos în sus a arborelui de derivare, respectiv în plasarea formulei ce descrie teza în poziția de secvent terminal și în reducerea acestuia prin aplicarea regulilor logice la secvenți inițiali ce satisfac exigențele unor axiome.

Evident, putem avea şi derivări ale unor secvenţi terminali din secvenţi iniţiali care nu sunt axiome. În acest caz, derivarea noastră va descrie doar faptul că adevărul secvenţilor iniţiali implică adevărul secventului terminal.

Derivarea în calculul secvenţial poate fi efectuată de la axiome la secventul terminal sau înainte (*forward reasoning*) sau invers, de la secventul terminal la formulele iniţiale (*backward reasoning*). Pe prima putem să o mai numim *inferenţă progresivă,* iar pe a doua *inferenţă regresivă* sau reducţie.

Inferenţa progresivă este o deducţie iar inferenţa regresivă este o întemeiere sau o argumentare.

2. Regulile calculului secvenţial

Regulile calculului secvenţial sunt de fapt scheme abstracte de derivare care devin propoziţii prescriptive dacă sunt interpretate adecvat.

În interpretarea asamblată pe care o folosim vom utiliza numai două tipuri de reguli: reguli iniţiale sau axiome şi reguli logice. Regulile structurale vor fi reduse la proprietăţi ale mulţimilor de formule.

Axiome sau reguli iniţiale

$\Gamma, A \rightarrow A, \Delta$ *schemă*

Reguli logice:

	st		**dr**	
		Negaţia		
(-st)	$\dfrac{\Gamma \rightarrow A, \Delta}{\Gamma, \neg A \rightarrow \Delta}$		$\dfrac{\Gamma, A \rightarrow \Delta}{\Gamma \rightarrow \neg A, \Delta}$	*(-dr)*
		Conjuncţia{tc "Conjuncţia"}		
(∧st)	$\dfrac{\Gamma, A, B \rightarrow \Delta}{\Gamma, A \wedge B \rightarrow \Delta}$	*(∧dr)*	$\dfrac{\Gamma \rightarrow A, \Delta \;;\; \Gamma \rightarrow B, \Delta}{\Gamma \rightarrow A \wedge B, \Delta}$	
		Disjuncţia{tc "Disjuncţia"}		
(∨st)	$\dfrac{\Gamma, A \rightarrow \Delta; \Gamma, B \rightarrow \Delta}{\Gamma, A \vee B \rightarrow \Delta}$		$\dfrac{\Gamma \rightarrow A, B, \Delta}{\Gamma \rightarrow A \vee B, \Delta}$	*(∨dr)*
		Implicaţia		
(⊃st)	$\dfrac{\Gamma \rightarrow A, \Delta \;;\; \Gamma, B \rightarrow \Delta}{\Gamma, A \supset B \rightarrow \Delta}$		$\dfrac{\Gamma, A \rightarrow B, \Delta}{\Gamma \rightarrow A \supset B, \Delta}$	*(⊃dr)*

Observații:

Simbolurile Γ și Δ nu intervin efectiv în calcul. Ele desemnează doar celelalte formule din antecedent și respectiv celelalte formule din succedent.

Formulele trecute în antecedent sunt considerate toate adevărate, iar cele din succedent, toate false. Importantă pentru aplicarea unei reguli este formula principală ce apare în antecedentul concluziei (pentru regulile marcate cu *st*) sau în succedentul concluziei (pentru regulile marcate cu *dr*).

Fiecare conectiv sau cuantificator logic are o regulă de introducere a sa în antecedent și o regulă de introducere a sa în succedent.

Citite de sus în jos, regulile descriu căi de agregare, compunere, sinteză sau construirea de formule complexe din cele atomare și, totodată, de deducere din „axiome" a „teoremelor" Parcurse în sens invers, de jos în sus, regulile descriu căi de analiză sau descompunere a agregatelor de formule în atomii constituenți și, totodată, de întemeiere sau argumentare a tezei.

Formula sau formulele scrise deasupra fiecărei linii reprezintă ipotezele sau premisele regulii, iar formula scrisă sub linie reprezintă concluzia.

Regulile din stânga *st* introduc un conectiv logic sau un cuantificator în antecedentul unui secvent; cele din dreapta, *dr.*, îl introduc și elimină din succedent. Formula ce conține acest conectiv sau cuantificator se numește *formula principală* și este subliniată în text.

Regulile marcate cu asterisc ∀dr și ∃st trebuie să satisfacă condițiile eigenvariabilelor, respectiv simbolurile ce desemnează parametri individuali nu trebuie să apară în concluziile regulilor.

Trecerea unei formule din antecedent în succedent (sau consecvent) sau invers se face prin schimbarea semnelor formulei (vezi regulile negației -st și -dr).

Toate regulile au un singur secvent în concluzie. Regulile logicii propoziționale se împart în α-reguli și β-reguli. Vom numi *α-reguli* sau de tip conjunctiv regulile care nu duc la scindarea arborelui de demonstrare, respectiv regulile care au un singur secvent și în premise. Vom numi *β-reguli* sau reguli de tip disjunctiv regulile care au în premise câte doi secvenți și, în consecință, duc la căi alternative în arborele de analiză. Vor fi β-reguli regulile *∧dr, ∨st, ⊃st*. Vom numi γ-reguli regulile ce introduc sau exclud cuantificatorii universali ∀st și

∃dr şi δ-reguli regulile ce introduc sau exclud cuantificatori existenţiali ∀dr şi ∃st.

Premisele unei reguli sunt alcătuite în exclusivitate din subformule ale concluziei regulii. Trecerea de la concluzie la premise într-o reducţie înseamnă în acelaşi timp analiză şi reducere la elementar.

Arborele de reducere generat prin aplicarea Bottom Up a regulilor logice la teza de demonstrat conduce la noduri etichetate prin formule ce au proprietatea de subformule.

Căutarea demonstraţiei sau construirea arborelui de demonstraţie în logica propoziţiilor

Să presupunem că ni se cere să demonstrăm prin calcul secvenţial o formulă din logica propoziţiilor, de exemplu, principiul tranzitivităţii.

Cum numărul axiomelor este infinit, este extrem de dificil să identificăm care sunt axiomele de la care trebuie să plecăm. Procedeul Top Down sau începerea demonstraţiei cu axiomele este neproductiv. (Există o metodă hibridă şi pentru această strategie. Identificăm FNC a expresiei. Apoi, din fiecare termen al FNC, respectiv din fiecare clauză disjunctivă identificăm axioma „ascunsă" în ea: variabilele negate scrise pozitiv vor alcătui antecedentul secventului, iar variabilele pozitive vor alcătui consecventul acestuia. Dar o astfel de procedură este pentru decizie inutilă, căci, din moment ce am adus formula la FNC, am şi aflat dacă ea este o tautologie sau nu. Metoda sugerată mai sus s-ar putea dovedi utilă pentru găsirea temeiurilor unei astfel de formule). În consecinţă, suntem constrânşi să utilizăm reductiv sau analitic formulele şi să apreciem tentativa noastră de a „demonstra" formula după cum reuşim sau nu să o reducem la secvenţi iniţiali axiome sau la secvenţi iniţiali infirmabili.

Dacă F este o formulă pe care vrem să o demonstrăm, atunci punctul de plecare al demersului nostru demonstrativ va fi secventul cu antecedentul vid, având drept succedent formula de demonstrat (∅, F), respectiv:

$$\rightarrow F \qquad \text{start}$$

Cu alte cuvinte, dacă F este o formulă validă, atunci ea trebuie să fie adevărată necondiţionat, respectiv să rezulte din mulţimea vidă de ipoteze. Analiza prin reguli logice se încheie atunci când secvenţii nu vor conţine decât atomi sau variabile şi, deci, nu mai putem aplica nici o regulă. Fie:

$$F = (p \supset q) \wedge (q \supset r) \supset (p \supset r)$$

Scriem, deci:

$$\rightarrow (p \supset q) \wedge (q \supset r) \supset (p \supset r) \qquad (1)$$

(1) este un secvent ce are o singură s-formulă şi anume o formulă succedent unică având drept semn principal $\supset$. Prin urmare, regula care se poate aplica pentru analiza ei este $\supset dr$.

Regula $\supset dr$ citită de sus în jos redă teorema deducției. Folosită ascendent, ea ne permite să trecem de la secventul (1) la un secvent (2) în care antecedentul implicației devine o formulă în antecedentul secventului şi consecventul implicației devine o formulă în succedentul secventului:

$$(p \supset q) \wedge (q \supset r) \rightarrow (p \supset r) \qquad (2)$$

Dacă convenim asupra citirii de jos în sus (mai întâi concluzia şi apoi premisele), atunci reducția va avea forma (3).

$$(p \supset q) \wedge (q \supset r) \rightarrow (p \supset r) \qquad \supset dr$$

$$\rightarrow (p \supset q) \wedge (q \supset r) \supset (p \supset r) \qquad (3)$$

Reducția (3) este o instanțiere a regulii $\supset dr$ pentru cazul când concluzia are forma secventului (1), respectiv $\Gamma = \varnothing$ şi $\Delta \neq \varnothing$.

Schema utilizată în reducția (3) spune că o formulă implicativă adevărată necondiționat (i.e. validă), poate fi redusă la un secvent potrivit căreia din antecedentul formulei se deduce consecventul ei. Simbolic acesta poate fi redat astfel:

$$\uparrow \frac{A \rightarrow B}{A \supset B}, \text{respectiv} \uparrow \frac{\Gamma, A \rightarrow B, \Delta}{\Gamma \rightarrow A \supset B, \Delta}$$

Săgeata $\uparrow$ vrea să indice că derivarea este de tip reductiv sau Bottom Up.

Citirea inversă a regulii $\supset dr$ (i.e. a teoremei deducției) ne permite să trecem ipotezele sau premisele ca antecedenți ai consecventului în succedentul secventului.

Putem spune că secventul (2) este temeiul imediat al lui (1), căci din (1), prin regula de inferență $\supset dr$ sau teorema deducției, aplicată Bottom Up, rezultă (2).

Analizăm mai departe secventul (2).

Acesta este un secvent având în antecedent o conjuncție şi în succedent o implicație. Avem acum de ales între două alternative: să aplicăm regula $\wedge st$ la formula moleculară din antecedent sau regula $\supset dr$ la implicația din succedent.

Aplicăm tot regula $\supset dr$ la implicaţia din succedentul secventului (2) şi scriem un nou secvent deasupra reducţiei (3), secventul numerotat prin (4)

$$(p\supset q) \wedge (q\supset r),p \rightarrow r \qquad (4)$$

Reducţiile efectuate până acum iau forma stivei de formule (5)

$$\begin{array}{ll} (p\supset q) \wedge (q\supset r),p \rightarrow r & \\ (p\supset q) \wedge (q\supset r) \rightarrow (p\supset r) & \supset dr \quad (5) \\ \rightarrow (p\supset q) \wedge (q\supset r) \supset (p\supset r) & \supset dr \end{array}$$

Continuăm analiza secventului din vârful stivei (secventul 4) cu ajutorul regulii $\wedge st$ şi obţinem secventul (6):

$$p\supset q, q\supset r, p \rightarrow r \qquad (6)$$

pe care-l scriem din nou în vârful stivei. Obţinem astfel o nouă configuraţie a stivei:

$$\begin{array}{ll} p\supset q, q\supset r, p \rightarrow r & \wedge st \\ (p\supset q) \wedge (q\supset r),p \rightarrow r & \\ (p\supset q) \wedge (q\supset r) \rightarrow (p\supset r) & \supset dr \quad (7) \\ \rightarrow (p\supset q) \wedge (q\supset r) \supset (p\supset r) & \supset dr \end{array}$$

Analiza secventului (6) ne lasă iar două alternative. Să aplicăm $\supset st$ la formula $p\supset q$ sau să aplicăm aceeaşi regulă la formula $q\supset r$.

Optăm pentru prima alternativă. Regula $\supset st$ este o β-regulă şi va genera o bifurcare. În prima, antecedentul implicaţiei analizate va apare ca o formulă derivată sau consecinţa din restul formulelor din succedentul secventului analizat şi în a doua consecventul formulei analizate va apare alături de celelalte formule ale antecedentului secventului.

Premisele rezultate din (6) prin aplicarea regulii $\supset$st sunt redate în (8):

$$(q\supset r),p \rightarrow p,r \qquad (8.1)$$
$$(q\supset r),p,q \rightarrow r \qquad (8.2)$$

Secventul din stânga, având formula p, atât în antecedent cât şi în succedent, este o axiomă. În consecinţă, aceasta ramificaţie se închide. Mai rămâne de analizat secventul 8.2. Formula principală este acum $q\supset r$ la care se aplică tot regula de analiză $\supset st$.

Din analiza ei obţinem premisele (8):

$$p,q \rightarrow q,r \qquad (8.2.1)$$
$$p,q,r\rightarrow r \qquad (8.2.2)$$

Întrucât ambele premise obţinute din 7.2 sunt axiome se închid şi aceste ramificaţii în arborele de analiză.

Reducerea prin reguli de analiză a formulei F a condus în toate ramificaţiile la frunze sau formule „iniţiale" ce sunt „axiome".

Putem acum, pornind de la aceste axiome, să obţinem prin derivări progresive sau inferenţe teza de demonstrat.

Redăm acum sintetic întregul demers reductiv şi deductiv, completând nivelurile superioare ale stivei 7.

(8.2.1) $p,q \rightarrow q,r$ (8.2.2) $p,q,r \rightarrow r$

(7.1) $(q \supset r),p \rightarrow p$ (7.2) $(q \supset r),p,q \rightarrow r$

6. $p \supset q,\ q \supset r,\ p \rightarrow r$

4. $(p \supset q) \wedge (q \supset r),p \rightarrow r$

2. $(p \supset q) \wedge (q \supset r) \rightarrow (p \supset r)$

1. $\rightarrow (p \supset q) \wedge (q \supset r) \supset (p \supset r)$

Structura de arbore de derivare o putem evidenţia mai bine prin marcare exclusivă a nodurilor şi regulilor utilizate.

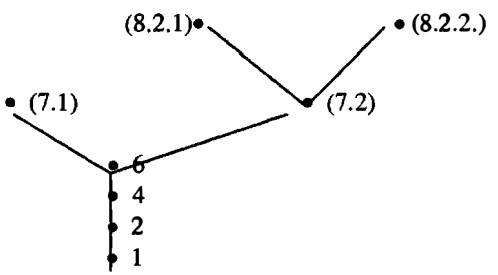


Fig 1 Arborele de derivare al principiului tranzitivităţii

Exemplul 2: Fie F1 = $((p \supset q) \wedge (r \supset q)) \supset ((p \vee r) \supset q)$

Verificarea validităţii unei formule prin calcul secvenţial:

\# \# 7.1. $r \rightarrow r, p, q$ 7.2. $r, q \rightarrow p, q$ \#

6.1. $r \supset q, p \rightarrow p, q$; 6.2. $r \supset q, p, q \rightarrow q$; 6.3. $r \supset q, r \rightarrow q, p$; 6.4. $r \supset q, r, q \rightarrow q$

5 1. $p \supset q, r \supset q, p \rightarrow q$; 5.2 $p \supset q, r \supset q, r \rightarrow q$ $\vee$st, 4

4. $p \supset q, r \supset q, p \vee r \rightarrow q$

3. $((p \supset q) \wedge (r \supset q)), p \vee r \rightarrow q$

2. $((p \supset q) \wedge (r \supset q)) \rightarrow (p \vee r) \supset q$

1. $\rightarrow ((p \supset q) \wedge (r \supset q)) \supset ((p \vee r) \supset q)$

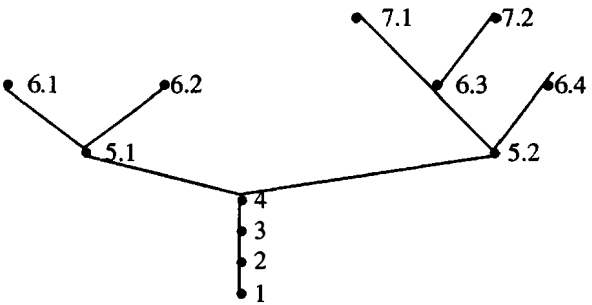


Fig.2. Arborele de derivare a formulei din exemplul 2

Întrucât formula F1 a putut fi redusă cu ajutorul regulilor de derivare $\supset dr$, $\wedge st$, $\supset st$ la formule iniţiale axiome, ea este validă. Pentru obţinerea deciziei au fost necesare 7 aplicaţii ale regulilor logice de reducere: $\supset dr{:}2$, $\wedge st{:}1$, $\supset st{:}3$. Acestea pot fi observate în arborele de derivare. Secvenţii 6.1, 6.2, 6.4, 7.1, 7.2 sunt secvenţi „iniţiali" sau axiome.

De fapt, întrucât noi am făcut o reducere, ei sunt secvenţi terminali. Ei devin secvenţi iniţiali şi axiome de îndată ce noi purcedem la demonstrarea progresivă a tezei aşezate la rădăcina arborelui.

Formula F2 este, de asemenea, validă.

Observaţii

Obs. 1 În aplicarea regulilor de reducţie am comprimat formulele ce conţineau subformule repetate prin formule fără repetare. Temeiul acesteia constă în legile de idempotenţă valabile, atât în logica propoziţiilor, cât şi în teoria mulţimilor.

Obs. 2 Formula F2 poate fi uşor modificată astfel încât să devină invalidă.

Recomandăm cititorului efectuarea unor astfel de modificări şi testarea prin calcul secvenţial a formulelor obţinute.

Observații finale despre calculul secvențial în logica propozițiilor

Calculul secvențial este o metodă eficace de decizie pentru formulele logicii propozițiilor.

Procedura de decizie constă în:

1. Plasăm formula de testat F în rădăcina viitorului arbore;

2. Aplicăm regulile calculului secvențial în sens reductiv, de jos în sus, până ce obținem *secvenți elementari* ce conțin numai literali, i.e. variabile, cu sau fără negație;

3. Secvenții elementari sau ireductibili, situați pe „frunze" pot fi axiome (și în acest caz $\Gamma \cap \Delta \neq \emptyset$) sau, dimpotrivă, secvenți infirmabili, respectiv secvenți care au contramodele;

4. O formulă este validă, dacă toți secvenții ei terminali (i.e. inițiali, dacă parcurgem arborele Top Down) sunt axiome și invalidă sau infirmabilă, dacă are cel puțin un secvent terminal (inițial) infirmabil.

Construirea contramodelului pentru un secvent terminal se face prin valorizarea cu 1 a formulelor din antecedent și valorizarea cu 0 a formulelor din succedentul secventului.

Regulile care conduc la bifurcații sau β-regulile sunt: $\vee$st, $\supset$st, $\wedge$dr. Toate celelalte sunt α-reguli și ele, aplicate Bottom Up, înlocuiesc o concluzie printr-o premisă unică a acesteia.

Calcularea secvenților este în logica propozițiilor întotdeauna finită. Dispunem de alternative în calcul, dar toate sunt proprii sau adecvate atingerii scopului. Admițând că *gradul* de complexitate al unui secvent este dat de numărul conectivelor logice din s-formulele sale, atunci aplicându-i acestuia în sens reductiv o regulă logică, vom obține întotdeauna premise cu un grad de complexitate mai scăzut. Orice proces de derivare va avea, în consecință, un număr finit de noduri etichetate prin secvenți și se va termina printr-un secvent elementar sau ireductibil.

Pentru decizia prin calcul secvențial a formulelor din logica propozițiilor *ordinea* aplicării regulilor de analiză nu are un rol esențial; ea poate întârzia puțin atingerea țelului. Cu totul altfel vor sta lucrurile în logica predicatelor.

Cap. 5. FORME NORMALE. DECIZIA PRIN FORME NORMALE

Formele normale sunt expresii standard cu structură fixă, care permit degajarea unor informaţii despre proprietăţile logice ale formulelor. De regulă, ele sunt construite pentru a decide asupra validităţii sau invalidităţii unei formule sau pentru descoperirea antecedenţilor sau consecinţelor acesteia. Mai pot fi folosite şi la verificare echivalenţei dintre două expresii.

Pentru a înţelege formele normale trebuie să reamintim, mai întâi, unele noţiuni preliminare.

Un *literal* este o variabilă sau o variabilă negată .

O conjuncţie elementară sau o *clauză conjunctivă* este o conjuncţie de literali:

$$c = \wedge \ell i, \infty > i \geq 1$$

unde ℓ este un literal iar i este un număr natural finit.

O *disjuncţie elementară* sau o *clauză disjunctivă* este o disjuncţie de literali:

$$d = \vee \ell i, i \geq 1$$

Numim *formă normală conjunctivă* o conjuncţie de disjuncţii elementare sau de clauze disjunctive.

$$F.N.C. = \wedge kj, j \geq 1, kj = \vee \ell i, i \geq 1$$

Analog, numim *formă normală disjunctivă* o disjuncţie de conjuncţii elementare, i.e.

$$F.N.D. = \vee cj, j \geq 1, cj = \wedge \ell i, i \geq 1$$

Exemple:

$p \wedge -q, p \wedge r \wedge -s$	sunt conjuncţii elementare
$-p \vee r, p \vee -q \vee r$	sunt disjuncţii elementare sau clauze disjunctive,
$(-p \vee r) \wedge (p \vee -q \vee r) \wedge (-s \vee t)$	este o formă normală conjunctivă, FNC.
$p\text{-}q \vee q\text{-}r \vee p\text{-}s$	este o formă normală disjunctivă, FND.

În ultimul exemplu am omis semnul conjuncţiei în termenii disjuncţiei, scriind *p-q* în loc de $p \wedge -q$. De asemenea, putem scrie semnul negaţiei deasupra variabilei în loc de a-l scrie ca un prefix , cum am făcut mai sus .

Principalele probleme pe care le vom urmări sunt: introducerea echivalenţelor utilizate în normalizarea expresiilor logic-propoziţionale, tipologia formelor normale şi decizia cu ajutorul formelor normale, formele normale şi aflarea consecinţelor şi ipotezelor unei formule, formele normale şi demonstraţia automatizată, formele normale, teoria modelelor şi metodologia sistemică.

1. Echivalenţele logic-propoziţionale şi normalizarea

Două formule A(p...,t) şi B(p...,t) sunt echivalente, dacă şi numai dacă, iau în orice atribuire de valori acordată variabilelor lor aceeaşi valoare de adevăr, i.e. desemnează aceeaşi funcţie de adevăr.

Echivalenţa expresiilor întemeiază regula substituirii echivalentelor denumită uneori şi *regula extensionalităţii, RE*

RE. Dacă într-o expresie E apare o formulă A şi A este echivalentă cu B, atunci, substituind în E pe A prin B (într-un caz sau mai multe de apariţie a lui A), obţinem o expresie E' echivalentă cu expresia iniţială E

Substituirea într-o formulă dată a unei expresii printr-o echivalentă a ei conservă valoarea de adevăr a formulei iniţiale. Acesta este şi temeiul pentru care regula *RE* se foloseşte în procesul „ normalizării „expresiilor

Redăm mai jos un număr de echivalenţe logice utilizate în aducerea unei formule la o formă normală conjunctivă sau disjunctivă:

15.1.1. $(p \oplus q) \equiv (p \vee q) \wedge (-p \vee -q)$

15.1.2. $(p \oplus q) \equiv (p \wedge -q) \vee (-p \wedge q)$

15.1.3. $(p \equiv q) \equiv (-p \vee q) \wedge (-q \vee p)$

15.14. $(p \equiv q) \equiv (p \wedge q) \vee (-p \wedge -q)$

15.17 $p \downarrow q \equiv -p \wedge -q$

15.1.5. $(p \rightarrow q) \equiv -p \vee q$

15.1.6. $p / q \equiv -p \vee -q$

15.1.8. $p [q \equiv p \wedge -q$

15.1.9. $p] q \equiv -p \wedge q$

15.1.10. $p \wedge (q \wedge r) \equiv (p \wedge q) \wedge r \equiv p \wedge q \wedge r$

15.1.11. $p \vee (q \vee r) \equiv (p \vee q) \vee r \equiv p \vee q \vee r$

15.1.12. $p \wedge (q \vee r) \equiv (p \wedge q) \vee (p \wedge r)$

15.1.13. $p \vee (q \wedge r) \equiv (p \vee q) \wedge (p \vee r)$

15.1.14. $p \vee q \equiv q \vee p$

15.1.15. $p \wedge q \equiv q \wedge p$

15.1.16. $- - p \equiv p$

15.1.17. $-(p \wedge q) \equiv -p \vee -q$

15.1.18. $-(p \vee q) \equiv -p \wedge -q$

15.1.19. $p \vee -p \equiv 1$

15.1.20. $p \wedge -p \equiv 0$

15.1.21. $p \vee 1 \equiv 1$

15.1.22. $p \wedge 1 \equiv p$

15.1.23. $p \vee 0 \equiv p$

15.1.24. $p \wedge 0 \equiv 0$

15.1.25. $p \vee (p \wedge q) \equiv p$

15.1.26. $p \wedge (p \vee q) \equiv p$

15.1.27. $p \wedge p \equiv p$

15.1.28. $p \vee p \equiv p$

Echivalenţele 15.1.1 – 15.1.9 sunt utilizate la eliminarea conectivelor nebooleene, iar echivalenţele 15.1.10 – 15.1.23, la prelucrarea expresiilor propoziţionale booleene.

2. Formele normale conjunctive

Fiind dată o formulă logică nebooleană A(p,q,...,t) putem construi o formulă echivalentă cu ea A'(p,q,...,t) în formă normală conjunctivă, exclusiv prin utilizarea regulii substituirii echivalentelor *RE* şi a echivalenţelor 15.11 – 15.1.28 enumerate mai sus.

Etapele principale ale aducerii unei formule în forma sa normală conjunctivă, FNC, sunt :

1. Eliminarea conectivelor nebooleene prin utilizarea regulii *RE* şi a echivalenţelor 15.1.1 – 15.1.9;
2. Eliminarea negaţiei expresiilor booleene compuse, prin utilizarea regulii RE şi a echivalenţelor 15.1.17 şi 15.1.18 .
3. Eliminarea semnelor de grupare inutile prin utilizarea legilor asociativităţii (15.1.10 şi 15.1.11);
4. Aplicarea legii de distributivitate 15.1.13 pentru a reduce componentele formulei la conjuncţii de disjuncţii elementare ;
5. Eliminarea termenilor care se repetă prin utilizarea legilor idempotenţei (15.1.27, 15.1.28).

O formulă adusă la FNC este lege logică, dacă şi numai dacă conţine o tautologie (o expresie de forma antecedentului lui 15.1.19) în fiecare termen al conjuncţiei, i.e. în fiecare clauză disjunctivă. O formulă adusă la FNC, ce nu conţine în fiecare clauză disjunctivă o subformulă de forma $A \vee -A$, este infirmabilă. Clauzele disjunctive netautologice ne arată în câte feluri putem infirma formula în cauză.

Exemple :

a) $((p \vee q) \rightarrow r) \rightarrow ((p \wedge -r) \rightarrow (q \rightarrow r))^*$

1.$((p \vee q) \rightarrow r) \rightarrow ((p \wedge -r) \rightarrow (q \rightarrow r))$

2.$((p \vee q) \wedge -r) \vee -(p \wedge -r) \vee -q \vee r$

3.$((p \vee q) \wedge -r) \vee -p \vee r \vee -q \vee r$

4.$(-r \vee -p \vee r \vee -q \vee r) \wedge (p \vee q \vee -p \vee r \vee -q \vee r)$

5.$(-r \vee -p \vee r \vee -q) \wedge (p \vee q \vee -p \vee r \vee -q)$

---

* Simbolul ' $\rightarrow$ ' utilizat în formula a) este simbolul implicaţiei . Noi folosim în mod curent pentru implicaţie simbolul ' $\supset$ ', deoarece simbolul ' $\rightarrow$' l-am folosit pentru relaţia de consecinţă logică din calculul secvenţial.

Formula a) este o lege logică sau o tautologie. Ea spune ca dacă p sau q implică pe r şi p nu implică pe r, atunci q implică r. în trecerea de la 1 la 2 am calculat mai întâi negația lui $(p \vee q) \supset r$, scriind $(p \vee q) \wedge -r$ şi am aplicat după aceea RE, utilizând echivalențele 15.1.5 şi 15.1.11, respectiv legea eliminării implicației şi legea asociativității disjuncției, punând în locul lui (p∧-r) ⊃(q⊃r) pe -(p∧-r) ∨ (q⊃r). în trecerea de la 2 la 3 am aplicat legea lui de Morgan la subformulă $-(p \wedge -r)$ pe care am scris-o mai jos ca $-p \vee r$ şi legea dublei negații 15.1.14, scriind pe --r ca r şi din nou legea asociativității disjuncției În trecerea de la 3 la 4 am aplicat legea distributivității disjuncției față de conjuncție, 15.1.13, luând drept p pe $-p \vee r \vee -q \vee r$, drept q pe $p \vee q$ şi drept r pe -r. În ultimul pas, am aplicat legea idempotenței disjuncției (15.1.28).

b) [(p ⊕ q) ∧ r] ⊃ [r ⊃ (-p ∧ s)]

1.[(p⊕q) ∧ r] ⊃ [r ⊃ (−p ∧ s)]

2 [(p ≡ q) ∨ −r] ∨ −r ∨ (−p∧s)

3 {[(−p ∨ q) ∧ (−q ∨ p)] ∨ −r} ∨ [(−r ∨ −p) ∧ (−r ∨ s)]

4.[(−p ∨ q ∨ −r) ∧ (p ∨ −q ∨ −r)] ∨ [(−p ∨ −r) ∧ (−r ∨ s)]

5 {[(−p ∨ q ∨ −r) ∧ (p ∨ −q ∨ −r) ∨ (−p ∨ −r)]} ∧ {[(−p ∨ q ∨ −r) ∧ (p ∨ −q ∨ −r)] ∨ (−r ∨ s)}

6.(−p ∨ q ∨ r ∨ −r) ∧ (p ∨ −q ∨ −r ∨ −p ∨ −r) ∧ (−p ∨ q ∨ −r ∨ −r ∨ s) ∧ (p ∨ −q ∨ −r ∨ s)

7. (−p ∨ q ∨-r ∨ s) ∧ (p ∨ -q ∨-r ∨ s)

Pentru trecerea de la 6 la 7 s-au utilizat regula extensionalității, RE (vezi pag 102) şi echivalențele 15.1.19, 15.1.21 şi 15.1.22. Formula b) nu este lege logică , deoarece 7 este infirmabilă

Formula b) este infirmabilă pentru p = r =1 şi q = s = 0 şi pentru p = s = 0 şi q = r = 1.

Într-adevăr, dând în b) variabilelor valorile: p = s = 0 şi q = r = 1 obținem:

$$[(0 \oplus 1) \wedge 1] \supset [1 \supset (1 \wedge 0)]$$

Pentru antecedent obținem: $(0 \oplus 1) = 1, 1 \wedge 1 = 1$

Pentru consecvent obținem pe rând, $1 \supset (1 \wedge 0), 1 \supset 0, 0$. Deci b) se reduce la, $1 \supset 0$, i.e la 0. Având o interpretare ce o infirmă, formula b) este invalidă.

3. Formele normale disjunctive

Orice formulă realizabilă poate fi adusă la o formă normală disjunctivă (FND).

Pentru a aduce o formulă la o formă normală disjunctivă trebuie, ca şi în cazul formelor normale conjunctive, să urmărim instrucţiunile 1 -- 3 i.e. să eliminăm operatorii nebooleeni, să coborâm negaţiile pe variabile prin utilizarea legilor lui De Morgan şi să aplicăm legile asociativităţii şi idempotenţei.

Pasul caracteristic va consta în utilizarea legii de distributivitate 15.1.12, care duce la proliferarea disjuncţiilor de conjuncţii elementare.

Pentru eliminarea termenilor care se repetă se utilizează, dacă e cazul, legile idempotenţei, 15.1.27 - 15.1.28.

Pentru ordonarea alfabetică a literalilor se utilizează legile comutativităţii 15.1.14 , 15.1.15.

O formulă adusă la FND este *realizabilă,* dacă şi numai dacă , are cel puţin un termen, i.e. are cel puţin o conjuncţie elementară. Fiecare termen indică o interpretare ce face formula iniţială adevărată.

Contradicţiile nu au forme normale disjunctive. Normalizarea disjunctivă a unei contradicţii duce la termeni incompatibili .

Exemple :

a) $((p \wedge q) \supset r) \supset (p \supset (q \supset r))$

1. $((p \wedge q) \supset r) \supset (p \supset (q \supset r))$

2. $-(-(p \wedge q) \vee r) \vee (-p \vee (-q \vee r))$ __ (RE,1,15.1.5)

3. $(- - (p \wedge q) \wedge -r) \vee (-p \vee (-q \vee r))$ __ (RE,2,15.1.18)

4. $((p \wedge q) \wedge -r) \vee (-p \vee (-q \vee r))$ __ (RE,3,15.1.16)

5. $p \wedge q \wedge -r \vee -p \vee -q \vee r$ __ (RE,4,15.1.11)

Trecerea de la 1 la 2 s-a făcut prin aplicarea regulii substituirii echivalentelor; peste tot în locul implicaţiei s-a pus, conform lui 15.1.5 negaţia antecedentului, disjunctă de consecvent.

Trecerea de la 2 la 3 s-a făcut tot pe baza regulii RE, dar utilizându-se, de data aceasta, o lege a lui De Morgan, 15.1.18.

Accesul la 4 a fost mediat de utilizarea legii dublei negaţii 15.1.16. În sfârşit, forma finală a fost atinsă prin utilizarea legilor de asociativitate, 16.1.10 şi 15.1.11.

FND ne sugerează interpretările care fac adevărată formula iniţială:

I1). r = 1 ; I2). q = 0 ; I3). p = 0; I4). p = q =1 şi r = 0

Să consideram primul caz: r = 1.

I1 a) $((p \wedge q) \supset 1) \supset (p \supset (q \supset 1))$

Dacă r = 1, atunci antecedentul lui a) va fi, potrivit definiției implicației, 1. Pe de altă parte, în consecventul lui a) dacă r = 1, $q \supset 1$, va fi 1 şi prin urmare $p \supset (q \supset 1)$ va fi 1, căci implicația este adevărată ori de câte ori are consecventul adevărat. în sfârşit, legând prin implicație, antecedentul care e 1 cu consecventul care e tot 1, obținem valoarea 1 pentru formula a) în interpretarea 1), r = 1. Analog, se raționează pentru celelalte 3 cazuri, I2-I4. FND face evidentă realizabilitatea sau irealizabilitatea unei formule.

b) $(p \supset q) \supset -(-q \supset -p)$

1.$(p \supset q) \wedge -(-q \supset -p)$

2.$(-p \vee q) \wedge -(--q \vee -p)$__(RE,1,15.1.5)

3.$(-p \vee q) \wedge -q \wedge p$__(RE,2,15.1.16,15.1.18)

4.$p \wedge -q \wedge q \vee p \wedge -q \wedge -p$__(RE,3,15.1.12)

5.0_ _ _ _(RE,4,15.1.20)

Exemplul b) ilustrează faptul că formulele irealizabile sau contradictorii; nu au FND.

c) $(p \vee -q) \wedge (q \supset r)$

1. $(p \vee -q) \wedge (q \supset r)$

2. $(p \vee -q) \wedge (-q \vee r)$

3. $((p \vee -q) \wedge -q) \vee ((p \vee -q) \wedge r)$

4. $-q \vee p \wedge -q \vee p \wedge r \vee -q \wedge r$

Formula c) este realizabilă. Pentru interpretarea q = 0 ea dă :

c) $(p \vee 1) \wedge (0 \supset r)$

1 ∧ 1

1

Vom conchide, subliniind faptul ca FND ne permite descoperirea tuturor interpretărilor în care formula analizată devine adevărată. FND nu ne spune explicit dacă formula cercetată este şi validă; ea ne atestă doar faptul că este realizabilă şi deci, nu este o contradicție.

Pentru verificarea validității trebuie să trecem de la FND la forma normală disjunctivă perfectă (FNDP).

4. Formele normale disjunctive perfecte

Particularitatea formelor normale disjunctive perfecte constă în faptul că identifică toate interpretările în care o formulă dată este adevărată şi o dată cu aceasta ne permite să ne pronunțăm asupra validității sau invalidității formulei. FNDP, ca şi metoda matriceală, ne furnizează o informație completă asupra veridicității unei formule date.

Numim *formă normală disjunctivă perfectă* a unei formule date o formulă echivalentă cu aceasta care este o forma normală disjunctivă, şi, în plus, fiecare termen al disjuncției conține toate variabilele de care depinde formula. Altfel spus, fiecare clauză conjunctivă conține toate variabilele ce apar în formulă.

Cum se construieşte o FNDP a unei formule?

Pentru a construi FNDP a unei formule urmărim următoarele instrucțiuni:

1. Construim mai întâi FND a formulei în cauză;

2. Ordonăm alfabetic (sau altfel) variabilele propoziționale care apar în formulă;

3. Descoperim care anume variabile propoziționale nu apar într-un termen sau altul al FND şi le introducem pe baza echivalențelor:

$$p \wedge 1 \equiv p \qquad (15.1.22)$$

$$p \vee -p \equiv 1 \qquad (15.1.19)$$

$$p \wedge (q \vee r) \equiv (p \wedge q) \vee (p \wedge r) \qquad (15.1.12)$$

Forma normală disjunctivă va fi disjuncția tuturor termenilor astfel încât fiecare să conțină toate variabilele. O formulă adusă la FNDP este lege logică sau formulă validă, dacă şi numai dacă are 2^n termeni sau constituenți de 1, unde n este numărul variabilelor care apar în formulă. Fiecare termen sau constituent de 1 desemnează un caz de adevăr al funcției asociate formulei inițiale sau o interpretare pentru care formula devine adevărată.

Exemple:

Să considerăm mai întâi formula a) din cap.15.3

a) $((p \wedge q) \supset r) \supset (p \supset (q \supset r))$

Forma normală disjunctivă a lui a) este expresia 5 (vezi cap. 15.3 exemplul a)

$$5. p \wedge q \wedge -r \vee -p \vee -q \vee r$$

a) este o funcţie φ de 3 variabile, $\varphi(p, q, r)$; deci, fiecare termen din 5 va trebui să conţină toate cele 3 variabile, fără însă ca introducerea acestora să-i schimbe valoarea de adevăr. Formula 5 are 4 termeni, primul termen, $p \wedge q \wedge -r$ conţine toate cele trei variabile; în consecinţă va intra neschimbat în FNDP; din cel de-al doilea termen, -p , lipsesc variabilele q si r ; din cel de-al treilea termen -q, lipsesc variabilele p si r; din cel de-al patrulea termen al FND, r, lipsesc variabilele p şi q. Introducem variabilele care lipsesc din ultimii trei termeni utilizând formulele, 15.1.22, 15.1.19 , 15.1.12 .

$$-p \equiv -p \wedge 1$$

$$-p \equiv -p \wedge (q \vee -q)$$

$$-p \equiv -p \wedge (q \vee -q) \equiv -p \wedge q \vee -p \wedge -q$$

Analog, vom avea:

$$-p \wedge q \equiv -p \wedge q \wedge r \vee -p \wedge q \wedge -r$$

$$-p \wedge -q \equiv -p \wedge -q \wedge r \vee -p \wedge -q \wedge -r$$

Deci:

$$-p \equiv -p \wedge q \wedge r \vee -p \wedge q \wedge -r \vee -p \wedge -q \wedge r \vee -p \wedge -q \wedge -r$$

Analog, vom calcula forma expandată a lui - q şi - r:

$$-q \equiv p \wedge -q \vee -p \wedge -q$$

$$p \wedge -q \equiv p \wedge -q \wedge r \vee p \wedge -q \wedge -r$$

$$p \wedge -q \equiv p \wedge -q \wedge r \vee p \wedge -q \wedge -r$$

de unde:

$$-q \equiv p \wedge -q \wedge r \vee p \wedge -q \wedge -r \vee -p \wedge -q \wedge r \vee -p \wedge -q \wedge -r$$

$$r \equiv q \wedge r \vee -q \wedge r$$

$$q \wedge r \equiv p \wedge q \wedge r \vee -p \wedge q \wedge r$$

$$-q \wedge r \equiv p \wedge -q \wedge r \vee -p \wedge -q \wedge r$$

$$r \equiv p \wedge q \wedge r \vee -p \wedge q \wedge r \vee p \wedge -q \wedge r \vee -p \wedge -q \wedge r$$

Putem acum rescrie expresia 5 ca:

5.
$p \wedge q \wedge \neg r \vee \neg p \wedge q \wedge r \vee \neg p \wedge q \wedge \neg r \vee \neg p \wedge \neg q \wedge r \vee \neg p \wedge \neg q \wedge \neg r$
$\vee p \wedge \neg q \wedge r \vee p \wedge \neg q \wedge \neg r \vee \neg p \wedge \neg q \wedge \neg r \vee p \wedge q \wedge r \vee \neg p \wedge q \wedge r \vee$
$p \wedge \neg q \wedge r \vee \neg p \wedge \neg q \wedge r$

de unde prin idempotență se obține:

7.
$p \wedge q \wedge r \vee p \wedge q \wedge \neg r \vee p \wedge \neg q \wedge r \vee p \wedge \neg q \wedge \neg r \vee \neg p \wedge q \wedge r \vee \neg p \wedge q \wedge \neg r \vee$
$\neg p \wedge \neg q \wedge r \vee \neg p \wedge \neg q \wedge \neg r$

Formula 7 este FNDP a formulei a). în cazul lui a), n=3, $2^3 = 8$. Formula 7 are 8 termeni sau constituenți de 1. Deci, a) este o for-mulă validă.

Considerăm acum formula:

c) $(p \vee \neg q) \wedge (q \supset r)$

Forma normală disjunctivă a lui c) este :

4) $p \wedge \neg q \vee \neg q \wedge \neg r \vee p \wedge r \vee \neg q \wedge r$

c) are trei variabile. Introducem variabilele care lipsesc din termenii FND.

$p \wedge \neg q = p \wedge \neg q \wedge r \vee p \wedge \neg q \wedge \neg r$
$\neg q \wedge \neg r = p \wedge \neg q \wedge \neg r \vee \neg p \wedge \neg q \wedge \neg r$
$p \wedge r = p \wedge q \wedge r \vee p \wedge \neg q \wedge r$
$\neg q \wedge r = p \wedge \neg q \wedge r \vee \neg p \wedge \neg q \wedge r$

Deci :

5) $p \wedge \neg q \wedge r \vee p \wedge \neg q \wedge \neg r \vee p \wedge \neg q \wedge \neg r \vee \neg p \wedge \neg q \wedge \neg r \vee p \wedge q \wedge r \vee p \wedge \neg q \wedge r \vee$
$p \wedge \neg q \wedge r \vee \neg p \wedge \neg q \wedge r$

6) $p \wedge \neg q \wedge r \vee p \wedge \neg q \wedge \neg r \vee \neg p \wedge \neg q \wedge \neg r \vee p \wedge q \wedge r \vee p \wedge \neg q \wedge r$

6) este FNDP a formulei c)

Observații:

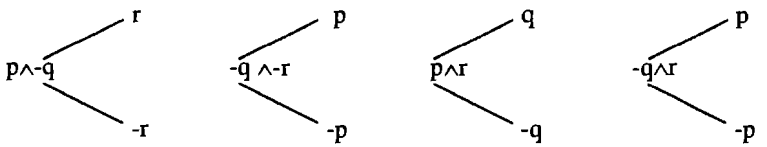
Obs. 1: Din 15.1.22. , 15.1.19 si 15.1.12 rezultă :

$$p \equiv p \wedge q \vee p \wedge \neg q$$
$$p \equiv p \wedge 1 = p \wedge (q \vee \neg q) = p \wedge q \vee p \wedge \neg q \qquad 15.4.1$$

Obs. 2: Putem conveni să omitem semnul conjuncţiei, pe $\wedge$ şi negaţia să o scriem deasupra variabilei. Atunci 15.4.1 va putea fi scrisă ca:

$$p = pq \vee p\,\overline{q} \qquad 15.4.2$$

Obs.3: Trecerea de la un termen al FND la constituenţi de 1 generaţi de acesta poate fi obţinută mult mai simplu cu ajutorul unei reprezentări grafice. Asociem fiecărui termen al FND, ce nu e în acelaşi timp un termen al FNDP, un arbore, plasând termenul în rădăcina acestuia şi adaugă succesiv variabile care lipsesc pe arce. Pentru FND a lui c) vom avea:



Identificarea constituenţilor generaţi de un termen al FND se face prin scrierea conjuncţiei tuturor literalilor din toate drumurile şi ordonarea alfabetică a acestora. Omiţând semnul conjuncţiei, obţinem FNDP a lui c), i.e. pentru 6) de mai sus, obţinem:

7) p-qr $\vee$ p-q-r $\vee$-p-q-r $\vee$ pqr $\vee$ p-qr

5. Formele normale conjunctive perfecte

Am văzut că FNDP ne enumeră toate interpretările sau valorizările pentru care o anumită formulă devine adevărată. Pot exista însă circumstanţe în care suntem interesaţi să cunoaştem toate interpretările în care o anumită formulă devine falsă. Pentru aceasta vom calcula forma normală conjunctivă perfectă.

Numim *formă normală conjunctivă perfectă* a unei formule date o formulă echivalentă cu aceasta care este o forma normală conjunctivă şi în plus fiecare termen al conjuncţiei conţine toate variabilele de care depinde formula.

Pentru a construi forma normală conjunctivă perfectă urmăm următoarele instrucţiuni :

1. Construim mai întâi FNC a formulei în cauză (vezi 22.15.2)
2. Identificăm numărul variabilelor distincte care apar în formulă şi le ordonăm alfabetic sau altfel;
3. Descoperim care anume variabile propoziţionale nu apar într-un termen sau altul al conjuncţiei şi le introducem în conformitate cu regulile:

$$p \vee 0 = p \qquad 15.5.1$$

$$p \wedge -p = 0 \qquad 15.5.2$$

$$p \vee (q \wedge r) = (p \vee q) \wedge (p \vee r) \qquad 15.5.3$$

Forma normală conjunctivă perfectă va fi conjuncția tuturor termenilor astfel îmbogățiți.

O formulă adusă la FNCP este *irealizabilă* sau *contradicție* daca are 2^n termeni sau constituenți de 0, unde n este numărul variabilelor care apar în ea .

Fiecare constituent de 0 desemnează un caz de infirmare a formulei.

Exemple:

Fie $\Psi = (p \vee q) \wedge (q \supset p) \wedge (p \supset r) \wedge (-p \supset -r)$

Aducem expresia Ψ la FNC:

1). $(p \vee q) \wedge (-q \vee p) \wedge (-p \vee r) \wedge (p \vee -r)$ FNC

In Ψ apar trei variabile distincte p, q, r. Trebuie, deci, ca fiecare termen al FNC să conțină toate cele trei variabile .

Din $p \vee q$ lipsește r. Îl introducem în concordanță cu 15.5.1. și 15.5.2.

$$p \vee q \vee 0 \equiv p \vee q \vee (r \wedge -r) \equiv p \vee q \vee r \wedge p \vee q \vee -r$$

Analog, pentru ceilalți trei termeni:

$$p \vee -q \equiv p \vee -q \vee 0 \equiv p \vee -q \vee (r \wedge -r) \equiv p \vee -q \vee r \wedge p \vee -q \vee -r$$

$$p \vee r \equiv p \vee 0 \vee r \equiv p \vee (q \wedge -q) \vee r \equiv p \vee q \vee r \wedge p \vee -q \vee r$$

$$p \vee -r \equiv p \vee 0 \vee -r \equiv p \vee (q \wedge -q) \vee -r \equiv p \vee q \vee -r \wedge p \vee -q \vee -r$$

Obținem astfel:

2) $(p \vee q \vee r) \wedge (p \vee q \vee -r) \wedge (p \vee -q \vee r) \wedge (p \vee -q \vee -r) \wedge$
$(-p \vee q \vee r) \wedge (-p \vee -q \vee r).$

2) este FNCP a formulei $\Psi(p,q,r)$. Întrucât nu are 2^3 constituenți de fals , Ψ nu este irealizabilă sau contradicție.

Ce putem, așadar, spune, în încheiere, despre formele normale și mai ales despre clasele de servicii pe care ni le pot aduce acestea.

Atât formele normale disjunctive (FND), cât și formele normale conjunctive (FNC) ne permit să punem în lumină ideile de *model* și *realizabilitate* a unui set de formule ce descriu o realitate fizică sau socială.

Fiecare termen al unei FND descrie un model sau interpretare adevărată pentru setul iniţial de formule.

Din orice FNC putem lesne construi un model prin punerea sub formă de listă a literalilor din clauzele unitare şi prin selectarea din fiecare clauză neunitară a câte unui literal, având grijă să nu trecem în listă vreo pereche opusă de literali.

Din fiecare clauză disjunctivă netautologică sau termen al formei normale conjunctive putem contrui un *contramodel* printr-o atribuire de valori care să facă clauza disjunctivă falsă. Aceasta va face forma normală conjunctivă falsă, căci o conjuncţie cu un termen fals va fi şi ea falsă. O formulă ce conţine un contramodel nu este o lege logică. Ea poate descrie însă o bază de cunoştinţe empirice sau un sistem de relaţii umane.

Formele normale conjunctive sunt un pas important în direcţia utilizării rezoluţiei Robinson şi în direcţia utilizării strategiilor rezolutive şi a demonstraţiilor automatizate. Tot pe formele normale conjunctive se bazează metodele de decizie Davis-Putnam şi metoda interpretărilor semantice succesive.

Formele normale disjunctive sunt un pas important în direcţia utilizării rezoluţiei duale sau a rezoluţiei Maslov.

6. Formele normale, teoria modelelor şi perspectiva sistemică

În rândurile de faţă dorim să facem o apropiere între teoria formelor normale şi perspectiva sistemică asupra prelucrării bazelor de cunoştinţe dintr-un anumit domeniu din realitate, în particular din teoria activităţilor umane.

Să considerăm, mai întâi, un set de formule care descrie o problemă de logică. O problemă este o bază de cunoştinţe şi un set de întrebări.

Din punct de vedere logic, o întrebare este o formulă logic propoziţională sau un predicat marcate de semnul interogaţiei.

Exemple: ?-p, ?-(p⊃q), ?- (r ∨ s ∨ t), ?- Q(x,y) i.e." Este adevărată p ?", „Este adevărat că p implică q?", „Este adevărat r sau s sau t ?", „Care sunt acele perechi de obiecte din domeniul D care satisfac relaţia $Q^{\#}$?"

Formalizăm în limbajul logicii propoziţiilor situaţia acţională descrisă în propoziţiile de la pag. 202.

1'. $p \supset q$

2'. $q \supset (r \vee s \vee t)$

3'. $q \supset (r1 \vee s1)$

4' $s1 \supset v1$

5'. $-v1$

6' p

6. $(s \vee t) \supset d$

7. -d

Formalizăm întrebările de la pag. 202:
a. ?- $(r \vee s \vee t)$; b. ?- $(r1 \vee s1)$

Aducem formulele de mai sus la formă clauzală, i.e. la FNC:

1" $-p \vee q$

2" $-q \vee r \vee s \vee t$

3" $-q \vee r1 \vee s1$

4" $-s1 \vee v1$

5" $-v1$

6" p

7.1" $-r \vee d$

7.2" $-s \vee d$

8 ". -d

Să considerăm forma normală conjunctivă:

$$S3 = (-p \vee q) \wedge (-q \vee r \vee s \vee t) \wedge (-q \vee r1 \vee s1) \wedge (-s1 \vee v1) \wedge -v1 \wedge p \wedge (-r \vee d) \wedge (-s \vee d) \wedge -d$$

S3 este o formă normală conjunctivă, i.e. o conjuncție de disjuncții elementare.

Observăm că S3 conține trei clauze unitare: p, -v1, –d şi şase clauze cu mai mulți literali, i.e. card(λ) >1, unde prin card(λ) am notat numărul literalilor dintr-o clauză. Acestea sunt:

$-p \vee q$, $-q \vee r \vee s \vee t$, $-q \vee r1 \vee s1$, $-s1 \vee v1$, $-r \vee d$, $-s \vee d$

Propunem următoarea interpretare sistemică a celor două tipuri de clauze:

1. Clauzele unitare descriu valorile de intrare în sistem;
2. Clauzele cu mai mulți literali descriu legile specifice sau regulile sistemului;

3. Aplicarea schemelor de inferenţă asupra intrărilor în sistem şi asupra legilor sau regulilor sistemului va genera valorile de ieşire. Valorile de ieşire vor fi consecinţe logice din clauzele de intrare şi din legile interne ale sistemului.

4. Lista clauzelor de intrare reprezintă modelul logic al *situaţiei iniţiale* în care se găseşte sistemul.

5. Clauzele unitare iniţiale (sau valorile de intrare), împreună cu clauzele unitare derivate (sau valorile de ieşire din sistem), alcătuiesc un *model* sau o interpretate adevărată pentru setul iniţial de clauze disjunctive sau pentru forma normală conjunctivă. Întrucât forma normală conjunctivă este logic echivalentă cu formula iniţială ce a fost adusă la forma normală conjunctivă, clauzele unitare iniţiale, împreună cu clauzele unitare derivate, vor fi un model al bazei de cunoştiţe sau al datelor unei probleme.

6. Clauzele unitare pot fi privite şi ca *model parţial* sau set de ipoteze iniţiale, care, admise împreună cu regulile sau clauzele generice ale domeniului de interpretare D, conduc la obţinerea *modelului complet* , model care include presupoziţiile latente sau consecinţele sistemului descoperite prin travaliul inferenţial. Astfel, logica poate fi înţeleasă ca o tehnică despre calcularea presupunerilor latente ce decurg din ipotezele explicite şi din regulile explicite ale domeniului de interpretare.

7. Condiţia declanşării unei noi consecinţe este includerea antecedentului unei reguli în lista de literali anterior deduşi, respectiv apariţia acestora ca literali negaţi în clauzele disjunctive.

8. Aflarea unui model complet înseamnă aflarea unei *soluţii* a problemei logice iniţiale.

9. Rezolvarea problemei logice poate fi făcută prin metodele calculului natural, prin metoda interpretărilor semantice succesive, prin metoda rezoluţiei, prin metoda Davis Putnam etc. A se vedea, în acest sens rezolvarea problemei redate formal mai sus, în capitolul 20, prin metoda interpretărilor semantice succesive.

10. Majoritatea problemelor formulate în limbile naturale, în limbajele teoriei mulţimilor, redate prin grafuri, arbori, grafice, etc pot fi redate prin formalizare în limbajul logicii predicatelor de ordinul întâi ca probleme logice.

Să cercetăm acum legătura dintre formele normale disjunctive şi perspectiva sistemică. Presupunem că am adus setul iniţial de formule la forma normală disjunctivă.

Aceasta ne oferă toate modelele sau interpretările ce satisfac problema logică în cauză.

Fiecare dintre termenii formei normale disjunctive va descrie o soluţie sau un model al problemei logice descrise de setul iniţial de formule. Dacă formula va avea un singură soluţie, atunci forma normală disjunctivă va avea un singur termen.

Forma normală disjunctivă a formulelor 1'- 7 din cap 6.9 scrise mai sus este:

S4 = pq-rr1-v1-d-s-s1t FND

Cititorul poate, desigur, calcula singur forma normală disjunctivă rezultată din formulele 1' – 7 enumerate mai sus.

Se observă că în S4 nu am mai scris semnul conjuncţiei între literali. Prin eliminarea semnului conjuncţiei, conform regulii excluderii conjuncţiei, obţinem:

{ p, q, -r, r1, -v1, -d, -s, -s1, t }, ceea ce coincide cu rezultatul obţinut prin formele normale conjunctive,

Int 9 = {p, q, -d, -s, -r, -v1, -s1, r1, t }.

Utilizarea FNC şi utilizarea FND pentru rezolvarea unei probleme logice a condus la acelaşi rezultat.

Interpretarea sistemică a formelor normale disjunctive propusă mai sus nu depăşeşte exigenţele logicii clasice în privinţa monotoniei. Ipotezele iniţiale nu se schimbă şi nici o consecinţă degajată din acestea nu-şi schimbă valoarea de adevăr.

Perspectiva sistemică propusă mai sus poate fi extinsă la o teorie logică nemonotonă, în care situaţiile iniţiale sau intrările în sistem vor putea fi diferite, în care baza iniţială de cunoştinţe va putea fi *extinsă, contractată şi revizuită.*

Vom încheia cu observaţia de interes metodologic şi tehnic aplicativ după care metoda formelor normale este o cale de rezolvare standardizată şi unitară a unei clase largi de probleme, din cele mai diverse domenii de aplicare, prin formalizare în limbajele logice şi prin decizia acestora prin tehnici şi metode logice variate şi eficace. Metoda formelor normale, ca o metodă algoritmică şi standardizată, este frecvent respinsă ca neintuitivă. Interpretarea semantico-sistemică propusă de noi mai sus o face îndeajuns de atractivă şi intuitivă. În plus, ea prezintă avantajul major că este, după crearea limbajelor de programare logică şi după descoperirea metodei rezoluţiei, pe deplin computerizabilă.

Cap. 6. METODA REZOLUŢIEI ÎN LOGICA PROPOZIŢIILOR

Vom prezenta, mai întâi, principiul rezoluţiei. Demonstrăm apoi în mai multe variante teorema rezolventului prin care arătăm că principiul rezoluţiei este o schemă validă de inferenţă. Pasul următor va consta în definirea conceptelor de derivare rezolutivă şi corespunzător de respingere rezolutivă. Primul concept va reface în limbajul teoriei rezoluţiei ideea de deducţie logică sau demonstraţie directă, iar cel de-al doilea pe cea de demonstraţie apagogică sau prin reducere la absurd. Teorema derivării rezolutive ne arata că orice aplicare iterată a principiului rezoluţiei la un set iniţial de clauze încheiată cu o clauză determinată echivalează cu demonstrarea acelei clauze.

Metoda rezoluţiei va fi prezentată ca o tehnică de testare a validităţii unor raţionamente, întemeiată pe aplicarea iterată a principiului rezoluţiei. Principiul rezoluţiei este, într-un anumit sens, o generalizare a schemelor clasice de inferenţă *modus ponens*, *modus tollens*, principiul silogismului ipotetic disjunctiv, principiul tranzitivităţii, principiul incompatibilităţii etc.

1. Schemele de inferenţă şi principiul rezoluţiei

O schemă de inferenţă este o regulă care garantează trecerea de la premise adevărate la consecinţe adevărate. Schemele de inferenţă înlesnesc pasul gândirii noastre discursive spre consecinţe mereu mai îndepărtate, în condiţiile conservării adevărului premi-selor în concluzii.

Iată câteva astfel de scheme de inferenţă:

	$A \supset B$		$A \supset B$		$A \vee B$
1)	A	2)	$-B$	3)	$-A$
	----		----		----
	....B.....		....–A.....		....B.....

4) $A \to B$ / $B \to C$ / ---- / $A \to C$

5) $-A \vee -B$ / A / ---- / ...$-B$...

Am enumerat, în ordine, *modus ponens, modus tollens*, principiul silogismului ipotetic – disjunctiv, legea tranzitivității și legea incompatibilității.

Aplicând echivalența 15.1.5, putem reformula regulile 1)--5) în termenii operațiilor logice booleene:

1° $-A \vee B$ / A / ----- / .....B....

2° $-A \vee B$ / $-B$ / ----- / ...$-A$....

3° $A \vee B$ / $-A$ / ----- / .....B.....

4° $-A \vee B$ / $-B \vee C$ / ----- / $-A \vee C$

5° $-A \vee -B$ / A / ----- / ....$-B$....

Astfel rescrise, schemele de inferență au ca „premise" și concluzii disjuncții elementare sau clauze disjunctive. În cazurile 1°, 2°, 3° și 5° avem cea de-a doua clauză și concluzia alcătuită din disjuncții elementare minime sau din clauze unitare, constituite dintr-un singur literal (un literal este, în logica propozițiilor, o variabilă propozițională în apariție simplă sau precedată de semnul negației).

Cazul 4° are, dimpotrivă, premisele și concluzia alcătuite din clauze disjunctive neunitare.

Observarea atentă a schemelor 1° – 5° ne permite să constatăm că, în trecerea de la premise la concluzii, s-a omis, în fiecare caz de inferență, o pereche de literali opuși -A și A, cazul 1°, B și -B în cazul 2° și așa mai departe, conectându-se prin disjuncție ceilalți literali rămași.

Așa, de exemplu, în cazul 4° perechea de literali opuși omiși este B și -B, iar disjuncția celor rămași este -A ∨ C. Redăm aceasta grafic prin:

$$\begin{array}{ll} & -A \vee B \\ 4 & -B \vee C \\ & \text{-----} \\ & -A \vee C \end{array}$$

Pentru a face şi mai transparent principiul utilizat în actul inferenţei rescriem asamblist schemele de inferenţă 1° – 5°. Fiecare clauză va fi scrisă ca o mulţime de literali:

$$\begin{array}{lll} \begin{array}{l} k_1:\{-A,B\} \\ k_2:\{A\} \\ \text{-------} \\ .....\{B\}..... \end{array} & \begin{array}{l} k_1:\{-A,B\} \\ k_2:\{-B\} \\ \text{-------} \\ ....\{-A\}..... \end{array} & \begin{array}{l} k_1:\{A,B\} \\ k_2:\{-A\} \\ \text{-------} \\ .....\{B\}..... \end{array} \\ 1' & 2' & 3' \end{array}$$

$$\begin{array}{ll} \begin{array}{l} k_1:\{-A,B\} \\ k_2:\{-B,C\} \\ \text{-------} \\ ..\{-A,C\}... \end{array} & \begin{array}{l} k_1:\{-A,-B\} \\ k_2:\{-A\} \\ \text{-------} \\ .....\{-B\}..... \end{array} \\ 4' & 5' \end{array}$$

Putem acum defini asamblist modul de obţinere a consecinţei sau rezolventului din cele doua premise transpuse sub forma unor clauze disjunctive. Astfel , pentru 1' vom avea:

$$res_A(k_1, k_2) = \{-A, B\} \setminus \{-A\} \cup \{A \setminus A\} = \{B\}$$

Analog, pentru 4' vom obţine:

$$res_B(k_1, k_2) = \{-A, B\} \setminus \{B\} \cup \{-B, C\} \setminus \{-B\} = \{-A, C\}$$

Notând prin k_1 şi k_2 cele două clauze şi prin l şi $-l$ perechea opusă de literali care apar în clauze, putem acum formula, într-o manieră generală principiul rezoluţiei: :

$$\begin{array}{ll} & \models k_1 \\ & \models k_2 \\ & l \in k_1 \\ & -l \in k_2 \\ (I) & \\ & res_l(k_1, k_2) = k_1 \setminus \{l\} \cup k_2 \setminus \{l\} \end{array}$$

Fiind date două clauze disjunctive k_1 şi k_2 astfel că prima conţine literalul l şi cea de-a doua literalul $-l$, putem scrie o nouă clauză, numită rezolventul lor în literalul l, obţinută prin reunirea

literalilor din prima clauză, din care s-a omis l, cu literalii din cea de-a doua clauză, din care s-a omis $-l$. Principiul rezoluției se aplică numai între clauze disjunctive ce conțin o pereche de literali opuși.

Exemplu 1:

$$\begin{array}{ll} k_1 = p \vee q, & k_1 = \{p,q\} \\ k_2 = -p \vee r & k_2 = \{-p,r\} \\ \hline res_p(k_1,k_2) = q \vee r & \{q,r\} \end{array}$$

Exempul 2: $res_p(p \vee q, -p \vee r) = \{p, q\} \setminus \{p\} \cup \{-p, r\} \setminus \{-p\} = \{q\} \cup \{r\} = \{q, r\}$

O formulare alternativă a principiului rezoluției este următoarea:

$$k_1 : A_1 \vee A_2 \vee \ldots \vee A_m$$

$$k_2 : B_1 \vee B_2 \vee \ldots \vee B_n$$

$$C : (\exists i)(\exists j)(1 \le i \le m, 1 \le j \le n, B_j = -A_i)$$

$$\text{---------- (II)}$$

$$res_{A_i}(k_1, k_2) = A_1 \vee A_2 \vee \ldots \vee A_{i-1} \vee A_{i+1} \vee \ldots$$

$$\ldots \vee A_n \vee B_1 \vee B_2 \vee \ldots \vee B_{j-1} \vee B_{j+1} \vee \ldots B_n$$

unde A_i și B_i sunt literalii opuși .

Condiția C cere ca în clauzele k_1 și k_2 să se găsească o pereche de literali opuși sau contradictorii

Este ușor de observat, că oricare dintre schemele 1° – 5 ° pot fi ușor obținute, prin degenerare, din principiul rezoluției formulat sub forma (II).

Pentru a obține, de exemplu, 1° din (II) substituim prin A pe fiecare dintre literalii clauzei k_1 și obținem $A \vee A \vee \quad \vee A$, care este echivalentă cu A în clauza k_2 substituim pe B_j prin -A și pe toți ceilalți literali $B_1, \ldots B_n$ prin B. Obținem astfel:

$$\begin{array}{l} A \\ -A \vee B \\ \hline \ldots\ldots B \ldots\ldots \end{array}$$

ceea ce este 1° cu clauzele inversate.

Pentru obținerea lui 4° din (II), substituim în k_1 pe fiecare dintre literalii $A_1, \ldots A_{m-1}$ prin -A și pe A_m prin B, iar în k_2

substituim pe B_1 prin -B şi pe fiecare dintre literalii $B_2 \ldots B_n$ prin C. Astfel obţinem:

$$k_1 : -A \vee -A \vee \ldots \vee B$$
$$k_2 : -B \vee C \vee \ldots \vee C$$
$$\text{---------------}$$
$$res_B(k_1, k_2) = -A \vee -A \vee \ldots C$$

de unde prin idempotenţă se obţine concluzia lui 4°.

Principiul rezoluţiei formulat sub forma II poate fi, în continuare, reformulat mai simplu.

Dacă vom conveni să notăm în II pe A_j prin l şi, corespunzător, pe $B_j = -A_1$ prin $-l$ şi să le scriem în clauzele k_1 şi respectiv k_2, atunci obţinem:

$$k_1 : l \vee A_1 \vee \ldots \vee A_m$$
$$\text{(III)} \quad k_2 : -l \vee B_1 \vee \ldots \vee B_n$$
$$\text{---------------------}$$
$$res_1(k_1, k_2) = A_1 \vee \ldots \vee A_m \vee B_1 \vee \ldots \vee B_n$$

În formularea (III) am putut omite condiţia stipulată în rândul trei din II, deoarece am făcut-o evidentă prin stipularea apariţiei lui l în k_1 şi a lui $-l$ în k_2

Putem comprima şi mai mult formularea principiului rezoluţiei introducând metavariabile pentru $A_1 \vee \quad \vee A_m$ şi pentru $B_1 \vee \quad \vee B_n$

$$k_1 : l \vee A$$
$$\text{(IV)} \quad k_1 : -l \vee B$$
$$\text{----------}$$
$$res_1(k_1, k_2) = A \vee B$$

A şi B vor denota în acest caz clauze disjunctive oarecare în care nu apar literalii l şi $-l$.

2. Teorema rezolventului

Până acum am prezentat principiul rezoluţiei ca o generalizare a schemelor de inferenţă *modus ponens*, *modus tollens*, principiul tranzitivităţii etc. şi am arătat că aceste scheme de inferenţă pot fi obţinute prin „degenerarea" principiului rezoluţiei. Acum dorim să

arătam că principiul rezoluției este o schemă de inferență validă, care conservă, în concluzie, „adevărul premiselor".

T2.1. Fie k_1 și k_2 două clauze astfel că $l \in k_1$ și $-l \in k_2$. Dacă este adevărată formula $k_1 \wedge k_2$, atunci va fi adevărată și rezolventul ei în l, $res_l(k_1,k_2)$.

Potrivit lui (I) rezolventul în l din k_1 si k_2 este definit ca:

$$\text{res}_l(k_1, k_2) = k_1 \setminus \{l\} \cup k_2 \setminus \{-l\}$$

Demonstrăm din ipoteze teorema rezolventului:

1. $k_1 \wedge k_2$ hyp
2. $l \in k_1$ hyp
3. $-l \in k_2$ hyp
4. k_1 E∧, 1.
5. k_2 E∧, 1.
6. $l = 1$ hyp. supl. 1
7. $-l = 0$ def. Neg 6
8. $k_2 \setminus \{-l\} = 1$, deoarece k_2 este 1 și $-l = 0$ (5,7)
9. $k_1 \setminus \{l\} \cup k_2 \setminus \{-l\}$, din 8 și def. lui ∪ și analog a lui ∨.
10. $res_l(k_1, k_2)$
11. $l = 0$ hyp. supl. 2
12. $k_1 \setminus \{1\} = 1$, deoarece k_1 este 1 și $l = 0$ (4,11)
13. $k_1 \setminus \{l\} \cup k_2 \setminus \{-l\}$. din 12 și def. lui ∪ și analog a lui ∨.
14. $res_l(k_1, k_2)$.

Să considerăm acum o altă cale de a demonstra teorema rezolventului. Dacă demonstrația de mai sus se întemeiază pe o interpretare asamblistă a clauzelor, cea pe care o propunem în continuare se întemeiază pe interpretarea exclusiv logic propozițională a clauzelor.

Pornim de la formularea (II) a principiului rezoluției:

Vom admite, deci, ca ipoteze:

1) $A_1 \vee A_2 \vee \ldots \vee A_m$

2) $B_1 \vee B_2 \vee \ldots \vee B_n$ } ip.

3) $(\exists i)(\exists j)(1 \le i \le m, 1 \le j \le n, B_j = -A_i)$

Fie i = a și j = b indicii pentru care 3) devine adevărată:

4) $B_b = -A_a$

Atunci 1) și 2) devin:

5)$A_1 \vee A_2 \vee A_a \vee ... \vee A_n$

6)$B_1 \vee B_2 \vee B_b \vee ... \vee B_n$

Deoarece are loc 4), putem înlocui în 6) pe B_b prin $-A_a$, obţinând astfel:

7)$B_1 \vee B_2 \vee -A_a \vee ... \vee B_n$

Admitem acum:

8)$A_a^v = 1$ ip. supl.

Atunci:

9)$-A_a^v = 0$

Pe aceasta bază 7) devine:

10)$B_1 \vee B_2 \vee ... \vee B_n$

căci $\alpha \vee 0 = \alpha$

Întrucât 10) este adevărată, pe baza lui $q \rightarrow (p \vee q)$, va fi adevărată :

11)$A_1 \vee A_2 \vee ... \vee A_m \vee B_1 \vee B_2 \vee ... \vee B_n$

Ori 11) este tocmai clauza rezolvent din 1) şi 2) în literalul A_a.

Admitem, acum, ipoteza suplimentară contrară lui 8):

12)$A_a^v = 0$ ip.supl.

Din 5), pe baza lui 12), rezultă

13)$A_1 \vee A_2 \vee ... \vee A_m$

De unde prin $p \supset (p \vee q)$ obţinem :

14)$A_1 \vee A_2 \vee ... \vee A_m \vee B_1 \vee B_2 \vee ... \vee B_n$

ceea ce coincide cu 11) şi este, din nou rezolventul în A_a din clauzele 1) şi 2), respectiv k_1, k_2 din formularea a doua a principiului rezoluţiei (II).

Astfel demonstraţia este completă.

Propunem şi o a treia demonstraţie a teoremei rezolventului. Pornim de data aceasta de la formularea a III-a a principiului rezoluţiei. În acest caz, clauzele sunt scrise mai comprimat prin utilizarea metavariabilelor A şi B ce ţin loc unor clauze oarecare în care nu apar literalii l şi $-l$.

Admitem, deci, definiţiile:

1)$k_1 = l \vee A$

2)$k_2 = -l \vee B$

3)$res_l(k_1, k_2) = A \vee B$

şi ipotezele:

4)k_1

5)k_2

la care adăugam ipoteza suplimentară:

6)$l^{\nu} = 0$

Pe baza lui 4) şi 5) şi a def.1) şi 2) putem scrie:

7)$l \vee A$

8)$-l \vee B$

Din 6) şi 7) rezultă prin $\alpha \vee 0 = \alpha$

9)A

De unde prin $p \supset p \vee q$ obţinem:

10)$A \vee B$

care, prin regula substituirii echivalentelor şi definiţia 3), duce la:

11) $res_l(k_1, k_2)$

ceea ce era de demonstrat .

Formulăm ipoteza suplimentară complementară:

12)$l^{\iota} = 1$

şi deci:

13)$-l^{\nu} = 0$

de unde din 8) şi $\alpha \vee 0 = \alpha$ rezultă :

14)B

iar din 14) prin $q \supset (p \vee q)$ se obţine :

15)$A \vee B$

din care prin regula extensionalităţii şi definiţia 3) rezultă:

16)$res_l(k_1, k_2)$,

ceea ce era de demonstrat.

Admiţând scrierea comprimată a principiului rezoluţiei (IV) putem reduce demonstrarea teoremei rezolventului la o aplicare banală a teoremei deducţiei şi la un exerciţiu la fel de banal de decizie a unei formule prin metoda arborilor de decizie .

Într-adevăr, teorema rezolventului cere ca fiind date două clauze k_1 şi k_2 astfel ca $l \in k_1$ şi $-l \in k_2$, din adevărul lor să se poată deduce adevărul clauzei rezolvent.

1) $k_1 \wedge k_2 \vdash res_l(k_1, k_2)$

respectiv:

2) $(l \vee A) \wedge (-l \vee B) \vdash A \vee B$

Prin teorema deducţiei 2) devine:

3)(l ∨ A) ∧ (–l ∨ B) ⊃ A ∨ B

Aplicând acum la 3) metoda arborilor de decizie obţinem:

1.{-[((*l*∨A) ∧(-*l* ∨ B)) ⊃(A∨B)]} ip.

2.{((*l*∨A) ∧(-*l* ∨ B), -(A∨B)} (-⊃)

3.{ (*l*∨A) ∧(-*l* ∨ B), -A, -B } (-∨)

4. {*l*∨A, -*l* ∨ B, -A, -B} (E ∧)

{*l*, -*l* ∨ B, -A, -B} 6. {A, -*l* ∨ B, -A, -B}
#

7. {*l*, -*l* , -A, -B} 8.{*l*, B, -A, -B}
# #

Întrucât s-au închis toate drumurile, formula 3), care descrie teorema rezolventului, s-a dovedit a fi validă .

Din demonstraţiile prezentate mai sus rezultă că principiul rezoluţiei este o schemă de inferenţă validă, care garantează adevărul clauzei rezolvent, dat fiind adevărul clauzelor „parentale", respectiv clauzelor premise .

Principiul rezoluţiei este o schemă de inferenţă generală care cuprinde ca nişte cazuri particulare schemele *modus ponens, modus tollens*, principiul silogismului ipotetic-disjunctiv, legea tranzitivităţii. Este o schemă de inferenţă standard ce utilizează, în exclusivitate, clauze disjunctive, îndeajuns de simplă pentru a putea fi computerizată.

3. Derivare rezolutivă şi respingere rezolutivă

Până acum am discutat despre raportul dintre o pereche de clauze şi un rezolvent ce rezultă din aceasta. Acum extindem discuţia noastră la lanţuri sau secvenţe de operaţii rezolutive .

D3.1 Fie *S* un set de clauze disjunctive şi *k* o clauză disjunctivă. Vom spune că *din S derivă rezolutiv k* şi vom nota $S \vdash_{res} k$, dacă şi numai dacă putem construi o secvenţă de clauze D în care fiecare element aparţine lui *S* sau dacă nu aparţine lui S, atunci există înaintea sa doua clauze din care aceasta rezultă prin rezoluţie.

Formal:

D3.1° $S \vdash_{res} k =_{dd}$ există o secvență

$D = \{k_1, \ldots k_n\}, k_n = k$

$(\forall i)\{[1 \leq i \leq n \supset k_i \in S] \vee [k_i \notin S \supset$

$\exists j,e,l(j,e < i \& res_l(k_j, k_e) = k_i)]\}$

A construi o derivare rezolutivă dintr-un set de clauze S pentru o clauză k înseamnă, așadar, a obține clauza în cauză după un număr finit de aplicări a principiului rezoluției la setul inițial de clauze. În secvența rezolutivă D, orice clauză sau aparține setului inițial de clauze S sau dacă nu aparține acestuia este derivată prin rezoluție din alte clauze anterioare ei. Secvența rezolutivă D se termină cu clauza k ce trebuie derivată din S.

A construi o derivare rezolutivă din S pentru k este, într-un fel, analog cu a construi pentru k un *text demonstrativ* dintr-un set de axiome S. Derivarea rezolutivă este o demonstrație standardizată ce folosește o singură regulă de inferență, principiul rezoluției.

Exemplu:

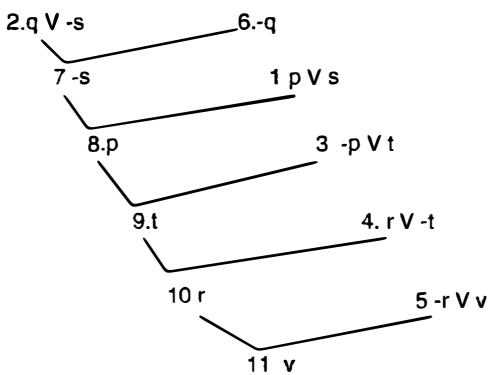
$$\begin{array}{cccccc} 1 & 2 & 3 & 4 & 5 & 6 \end{array}$$

$$S = \{p \vee s, q \vee -s, -p \vee t, r \vee -t, -r \vee v, -q\}$$

$k = v$

Se cere : $S \vdash_{res} v$

Rezolvare:



$$\begin{array}{cccccccccccc} 2 & 6 & 7 & 1 & 8 & 3 & 9 & 4 & 10 & 5 & 11 \end{array}$$

$$D = \{q \vee -s, -q, -s, p \vee s, p, -p \vee t, t, r \vee -t, r, -r \vee v, v\}$$

Este uşor de văzut că secvenţa D satisface condiţiile stipulate de definiţia 3.1; Clauzele 2 şi 6 sunt din S, setul de premise iniţiale; Clauza 7, -s, nu este din S, dar este derivată din 2 şi 6, care sunt anterioare lui 7; 1 este din S, 8 nu este din S, dar are alte două clauze anterioare din care derivă, 7 şi 1; 3 este din S, 9 nu este din S, dar are alte două clauze ce o preced din care derivă, 8 şi 3; 4 este din S. 10 nu este din S, dar are înaintea sa pe 9 şi pe 4, din care derivă. 5 este din S. 11 nu este din S dar derivă din alte două clauze anterioare, 10 şi 5.

Ultima clauză în secvenţa D este chiar clauza k, respectiv clauza dintr-un singur literal v, care trebuia derivată.

Este uşor de văzut că o derivare rezolutivă este o aplicare repetată a principiului rezoluţiei care se termină cu obţinerea clauzei căutate. Problema practică ce se pune acum este de a determina după ce criterii trebuie să alegem perechea de clauze cu care începem derivarea şi literalul în care facem rezoluţia, dacă între clauzele selectate sunt posibile mai multe aplicaţii ale principiului rezoluţiei în 2 sau mai mulţi literali. Dar aceste probleme ţin de rafinarea metodei rezoluţiei de care ne vom ocupa ulterior

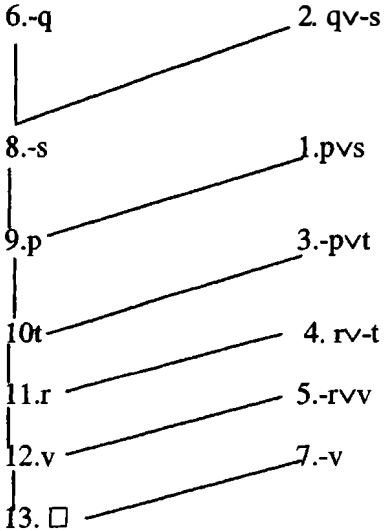
D3.2. Numim *respingere rezolutivă* a unui set de clauze S o derivare rezolutivă din S încheiată cu obţinerea unei contradicţii, i.e. a clauzei vide, $\square = p \wedge -p$.

Un set de clauze S din care am putut obţine clauza vidă este incompatibil. Din orice derivare rezolutivă putem obţine un set incompatibil de clauze prin acceptarea setului iniţial de clauze şi a negaţiei clauzei derivate. Aşa de exemplu, din setul $S = \{p \vee s, q \vee -s, -p \vee t, r \vee -t, -r \vee v, -q\}$ din care am derivat clauza v, putem forma setul incompatibil:

$$S' = S \cup \{-v\} = \{p \vee s, q \vee -s, -p \vee t, r \vee -t, -r \vee v, -q\} \cup \{-v\} =$$

$$= \{p \vee s, q \vee -s, -p \vee t, r \vee -t, -r \vee v, -q, -v\}$$

Să testăm acum incompatibilitatea lui S'.



Într-adevar, setul S' este incompatibil.

Un set incompatibil de clauze, ca S' de mai sus, conține potențial o mulțime de raționamente valide. Pornind de la el putem lesne formula mai multe raționamente valide.

dacă $S = \square$ atunci $-S = \blacksquare = T$

Cum însă $S = \wedge_{i\geq 1} k_i$, iar $k_i = \vee_{j\geq 1} l_j$,

$-S = -\wedge_{i\geq 1} k_i = \vee_{i\geq 1} -k_i$ și $-k_i = \wedge_{j\geq 1} -l_j$

$-S = \vee(\wedge_{i,j\geq 1} -l_j)_i = \blacksquare = T$

Dacă:

$S = S' = (p \vee s) \wedge (q \vee -s) \wedge (-p \vee t) \wedge (r \vee -t) \wedge (-r \vee v) \wedge -q \wedge -v = \square$

atunci:

$-S = -S = (-p \wedge -s) \vee (-q \wedge s) \vee (p \wedge -t) \vee (-r \wedge t) \vee (r \wedge -v) \vee q \vee v = \blacksquare$

Din orice set de clauze disjunctive incompatibile putem construi o tautologie în formă normală disjunctivă prin înlocuirea literalilor prin opușii lor și a operațiilor booleene prin dualele lor.

În mod analog, de la o formă normală disjunctivă a unei tautologii putem construi un set de clauze incompatibile.

Numărul scindărilor în două sau al dihotomiilor posibile cu cele n clauze este dat de:

$$N_{rv}=\overset{k=n}{\underset{k=0}{U}} C_n^k \qquad 3.3$$

iar numărul combinaţiilor de n obiecte luate câte k este dat de:

$$C_n^k=\frac{n!}{k!(n-k)!} \qquad 3.4$$

Fie $S=(k_1 \wedge ... \wedge k_2)$, deci n = 4. Atunci numărul raţionamentelor valide va fi:

$$N_{rv}=\overset{k=4}{\underset{k=0}{U}} C_4^k = C_4^0+C_4^1+C_4^2+C_4^3+C_4^4=$$
$$=1+4+6+4+1=16$$

Aşadar, din n clauze incompatibile, putem genera 2^n raţionamente valide.

4. Teorema derivării rezolutive

Problema care ne-o punem în acest moment este cea a raportului dintre o derivare rezolutivă a unei clauze k dintru-un set de clauze S, şi cea a deducerii dintr-un set de premise S a unei consecinţe k.

Răspunsul corect la această întrebare este dat de teorema derivării rezolutive.

T4.1. *Dacă dintr-un set de clauze S derivă rezolutiv clauza k, atunci din S se deduce clauza k.*

Simbolic: Dacă $S \models_{res} k$, *atunci* $S \models k$.

Demonstraţie.

Admitem adevărul lui S şi derivarea rezolutivă a lui k din S, respectiv:

1) S ip.

2) $S \models_{res} k$ ip.

Dacă clauza k este, conform ipotezei 2, o derivare rezolutivă din S, atunci există o secvenţă de clauze:

3) $D= \{k_1, k_2, ..., k_n\}$

astfel că:

4) $k_n = k$

şi

5) Pentru orice k_j din D, cu $1 \le j \le n$, $k_j \in S$ sau dacă $k_j \notin S$, atunci există în D două clauze k_e şi k_h, cu e,h<j şi există un literal *l* astfel că $k_j = res_l(k_e, k_h)$

Vom construi o demonstrație prin inducție asupra lungimii secvenței de clauze D = [k_1, k_2, ..., k_n], arătând că, dacă este adevărată prima clauză rezolvent din D, i.e. P(1) și dacă adevărul unei clauze k_i din D implică în mod necesar adevărul clauzei imediat succesive k_{i+1}, atunci va fi adevărată *orice* clauză k_i din D, inclusiv k_n sau k.

Observăm, mai întâi, că primele două clauze din D, k_1 și k_2, trebuie să aparțină lui S și, deci, potrivit ipotezei 1), sunt adevărate, căci altfel ar însemna că există înaintea lor alte clauze din care ar fi derivate rezolutiv. Deci, vom scrie:

6) k_1

7) k_2

Clauza k_3 este prima clauză rezolvent și a fost obținută prin rezoluție din k_1 și k2. Potrivit teoremei rezolventului (vezi **16.2**), k_3 din D este adevărată:

8) k_3

Am demonstrat, deci, P(1), i. e., P(k_3)

Fie acum un *i* fixat astfel că pentru orice $j \leq i$, k_j (indicele inferior este j) este adevărat:

9) k_j, $\forall j \leq i$

Atunci k_{i+1} va fi și ea în mod necesar adevărată. Într-adevăr, k_{i+1} poate fi:

o clauză din S, k_j, cu $j \leq i$ și atunci, potrivit ipotezei 1), k_{i+1} va fi adevărată, căci am admis că toate clauzele inițiale sunt adevărate;

k_{i+1} este o clauză rezolvent obținută din alte două clauze anterioare lui k_i. Dar am admis, puțin mai sus (vezi 9)) că toate clauzele anterioare lui k_i sunt adevărate, respectiv orice clauză k_j, cu $j \leq i$, este adevărată.

Din nou prin teorema rezolventului vom conchide asupra adevărului lui k_{j+1}

Am demonstrat astfel: P(i)→ P(i+1).

Având demonstrate P(1) și P(i)→ P(i+1), condițiile raționamentului prin inducție sunt satisfăcute. În consecință, pentru orice m, $1 \leq m \leq n$, P(m), i.e. orice clauză k_m va fi adevărată, inclusiv când m = n, $k_m = k_n$. Dar k_n este tot una cu k, clauza derivată din S (vezi 4).

Am demonstrat, deci, că din S se deduce k, i.e., S ⊨ k .

5. Principiul rezoluţiei şi metoda rezoluţiei

O metodă este un ansamblu de reguli sau instrucţiuni prin care se rezolvă o clasă de probleme. Aplicarea unei metode se reduce la identificarea unor stări ale sistemelor şi la aplicarea instrucţiunilor prevăzute în metodă pentru acele stări. Aplicarea unei metode este o conduită teleologică călăuzită de reguli. Scopul este rezolvarea problemei.

Sunt metode algoritmice şi metode nealgoritmice. Metodele algoritmice servesc la introducerea calculatorului şi a automatizării în rezolvarea problemelor.

Să presupunem că sunt date un număr de raţionamente formalizabile în limbajul logicii propoziţiilor. Care vor fi, în acest caz, principalele momente pentru a le rezolva prin metoda rezoluţiei? Principalele momente vor fi:

1. formalizarea raţionamentului în limbajul logicii propoziţiilor;
2. aducerea premiselor raţionamentului la formă clauzală disjunctivă;
3. negarea concluziei şi aducerea la formă clauzală a negaţiei concluziei;
4. formarea setului S de clauze incompatibile al raţionamentului prin conjugarea clauzelor obţinute din premise cu cele obţinute din negarea concluziei;
5. construirea unei respingeri rezolutive pentru setul de clauze S.

Dacă respingerea rezolutivă reuşeşte, i.e. se obţine clauza vidă, atunci raţionamentul iniţial este valid.

Metoda rezoluţiei poate fi redată concis [15, 163-4] prin secvenţa de instrucţiuni.

CLAUZE ← S

până ce □ *aparţine mulţimii* CLAUZELOR *continuă să faci* (do)

începe (begin)

alege din CLAUZE *două clauze distincte rezolubile* c_i *şi* c_j

calculează un rezolvent r_{ij} *din clauzele* c_i *şi* c_j

CLAUZE ← *Mulţimea obţinută prin adăugarea rezolventului* r_{ij} *la* CLAUZE

Sfârşeşte (end).

Postulăm, deci, pe S drept set inițial de clauze. Alegem din S, după o anumită strategie, două clauze rezolubile într-un literal oarecare și calculăm rezolventul lor.

Îmbogățim setul inițial de clauze cu rezolventul obținut și continuăm procesul selectării câte unei perechi de clauze, calculării rezolventului lor și adăugirii lui la setul anterior de clauze până când obținem, după un număr finit de iterări, clauza vidă.

Metoda rezoluției este, așa dar, o aplicare repetată a principiului rezoluției, în condițiile extinderii permanente a setului inițial de clauze S prin adăugarea rezolvenților obținuți.

Partea mai subtilă a metodei constă în identificarea criteriilor după care alegem perechea de clauze următoare la care aplicăm principiul rezoluției ca unică schemă de inferență.

Exemplu

1. Dacă Ștefan termină facultatea, va lucra în industria electronică sau în cercetare.
2. Dacă va lucra în industria electronică va controla execuția proceselor tehnologice.
3. Dacă va lucra în cercetare va face descoperiri sau invenții.
4. Dar Ștefan nu va controla execuția proceselor tehnologice și nu va inventa sau descoperi ceva.
5. Prin urmare, Ștefan nu va termina facultatea.

Formalizăm mai întâi raționamentul:

1. $p \supset (q \vee r)$
2. $q \supset s$
3. $r \supset (t \vee v)$
4. $\underline{-s \wedge -(t \vee v)}$
5. $-p$

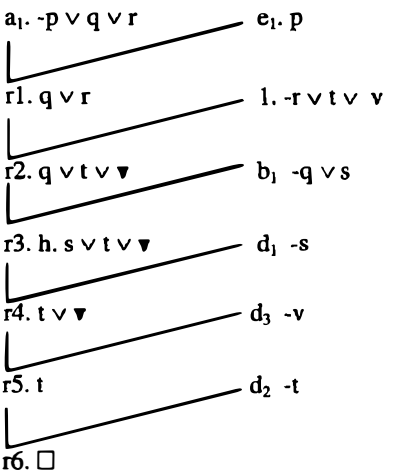
Transpunem premisele raționamentului sub forma unor clauze disjunctive.

a_1 $-p \vee q \vee r$
b_1 $-q \vee s$
c_1 $-r \vee t \vee v$
d_1 $-s$
d_2 $-t$
d_3 $-v$

Negăm concluzia și o transpunem ca o clauză disjunctivă: e_1 p.

Setul S de clauze incompatibile va fi alcătuit din: a_1, b_1, c_1, d_1, d_2, d_3 şi $-e_1$

Respingerea rezolutivă va putea avea forma:



Întrucât $\square \in S$, raţionamentul iniţial este valid.

6. Strategii rezolutive

Având un set de clauze incompatibile metoda respingerii rezolutive constă, aşadar, în selectarea în ordine a unei perechi de clauze rezolubile, în calcularea rezolventului lor şi în adăugarea acestuia la setul iniţial de clauze, până când obţinem clauza vidă. Problema care se pune este după ce criterii alegem perechile de clauze din care degajăm rezolventul, în câte feluri putem clasifica şi ordona setul iniţial de clauze incompatibile, după ce reguli ne călăuzim în producerea rezolvenţilor.

a. Strategia explorării în lăţime (the breadth first strategy)

Presupunem dat un set de clauze incompatibile S. Admitem că sunt posibile, la început, mai multe rezoluţii. Întrebarea care se pune în acest caz este următoarea: este mai avantajos să producem mai întâi toate inferenţele imediate din setul iniţial de clauze şi abia după aceea să trecem la inferenţe mai îndepărtate în care rezolvenţii apar ca premise sau, dimpotrivă, este mai avantajos să facem străpungeri în adâncime extrăgând rezolvenţii rezolvenţilor şi aşa mai departe.

Strategia analizei în lăţime ne cere să degajăm mai întâi toţi rezolvenţii imediaţi din setul iniţial de clauze şi abia după aceea, adăugându-i pe aceştia la setul anterior, să extragem un nou nivel de rezolvenţi şi aşa mai departe. Este un fel de metodă a sitei dese care scrutează atent fiecare pas imediat accesibil, dar care, din nefericire, înaintează greu către concluzia dorită.

Exemplu:

$S = (p \vee q) \wedge (-p \vee r) \wedge -r \wedge (-q \vee s) \wedge (-s \vee r)$
1 2 3 4 5

Extragem din S mulţimea rezolvenţilor imediaţi.

$R1 = \{ q \vee r, p \vee s, -p, -s, -q \vee r\}$
6 7 8 9 10

Menţionăm în paranteze clauzele–părinţi din care au fost obţinuţi rezolvenţii din R1: 6(1, 2), 7(1, 4), 8(2, 3), 9(5,3), 10(4, 5)

$S1 = S \cup R1$

$R2 = \{q, r \vee s, r \vee s, p \vee r, q, q, p \vee r, -q\}$
11 12 13 14 15 16 17 18

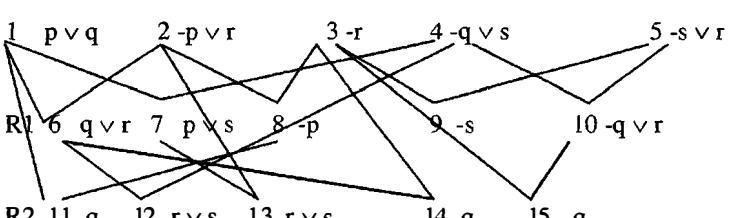
Rezolvenţii din R2 au provenit din clauzele-părinţi specificate în paranteze: 11(6, 3) sau 11(1,8),12(4, 6), 13(2, 7), 14(6, 3), 15(3,10) sau 15(8,1)sau 15(9,4), 17(10,1), 18(10, 3).

$S2 = S1 \cup R2$

Dar S2 este incompatibilă, căci conţine clauzele 11, 18, respectiv pe q şi -q. Pentru demonstrarea inconsistenţei lui S au fost necesare 11 aplicaţii ale principiului rezoluţiei (vezi obţinerea rezolvenţilor 6-10 din R1 şi 11–16 din R2).

Strategia urmată a fost în exemplul dat eficace, dar nu şi economicoasă.

Demersul inferenţial de mai sus poate fi redat grafic astfel:

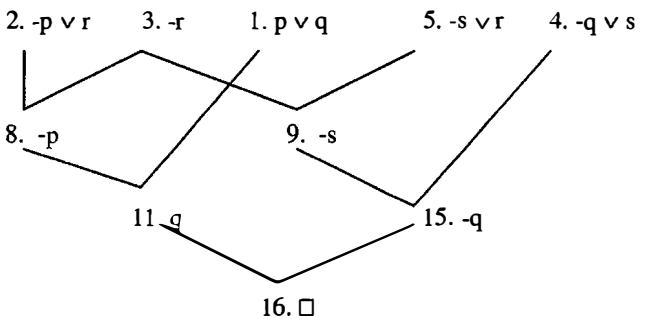


$S1 = S \cup R1$

$S2 = S1 \cup R2$

$S3 = \square$

Reluând numai inferențele strict necesare obținem:



În exemplul cercetat eficiența strategiei explorării în lățime a fost de 5/13, căci au fost realizate 13 acte inferențiale (18 - 5 = 13) în raport cu cele 5 strict necesare.

Mai trebuie reținut că strategia explorării în lățime nu garantează în toate cazurile captarea clauzei vide.

b. Strategia mulțimii de suport

Fie S un set incompatibil de clauze. Admitem că S se divide într-o submulțime S1 a clauzelor provenite din negația concluziei şi într-o submulțime S2 a clauzelor provenite din formalizarea premiselor unui raționament valid. Admitem, de asemenea, că S2 sau mulțimea premiselor are cel puțin o interpretare sau un model în care toate clauzele sunt adevărate. în acest caz, pentru a capta clauza vidă i.e. a obține incompatibilitatea, deoarece S2 este consistentă, va trebui în procesul rezolutiv să folosim în mod necesar clauze din S1 sau din aşa numita mulțime de suport.

În mod general, o mulțime de clauze S1 este *mulțime de suport*, dacă şi numai dacă există în S o mulțime S2 care are o interpretare sau un model şi S1 este diferența dintre S şi S2, i.e.S1 = S\S2. S1 este mulțimea de clauze ce nu satisface interpretarea I satisfăcută de premisele S2. Metoda strategiei de suport este un caz particular de rezoluție semantică.

Particularitatea *rezoluţiei semantice* este faptul că orice rezoluţie sau obţinere de rezolvent se face între o clauză din S1, mulţimea clauzelor ce nu satisfac interpretarea I şi alta din S2, mulţimea clauzelor ce satisfac interpretarea I şi că rezolventul obţinut, dacă satisface interpretarea I se include în S2, iar dacă nu o satisface, se include în S1. Metoda strategiei de suport este eficace şi eficientă.

Exemplul 1

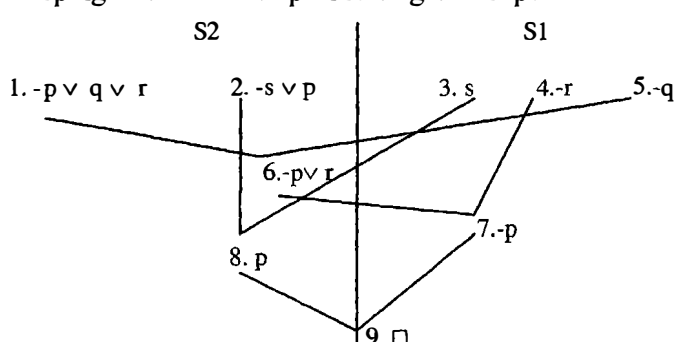
I Dacă oboseşti, te culci sau bei cafea.
Dacă munceşti, oboseşti.

Prin urmare, dacă munceşti şi nu bei cafea, te culci.

II $p \supset (q \vee r)$
$s \supset p$
$(s \wedge -r) \supset q$

III 1. $-p \vee q \vee r$ $I = \{p, q, r, -s\}$
2. $-s \vee p$ $S1 = \{s, -r, -q\}$
3. s $S2 = \{-p \vee q \vee r, -s \vee p\}$
4. $-r$
5. $-q$

Respingerea rezolutivă prin strategia de suport:

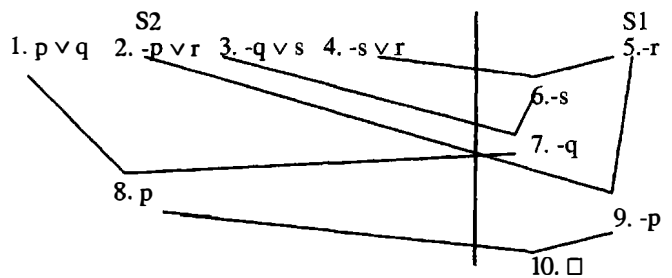


După cum este uşor de observat, orice rezoluție se face prin utilizarea unei clauze din S1 şi a alteia din S2; „părinții" au grupe sangvine diferite! Rezolvenții au fost scrişi în partea stângă sau în S2, dacă satisfac interpretarea I (a se vedea 6 şi 8) şi în partea dreaptă, dacă nu o satisfac (7 şi 9).

Exemplul 2

Fie $S = (p \vee q) \wedge (-p \vee r) \wedge -r \wedge (-q \vee s) \wedge (-s \vee r)$

Admitem $I = \{p, q, r, s\}$. În acest caz, $S1 = \{-r\}$ şi $S2 = \{p \vee q, -p \vee r, -q \vee s, -s \vee r\}$. iar respingerea rezolutivă întemeiată pe metoda strategiei mulțimii de suport va fi:



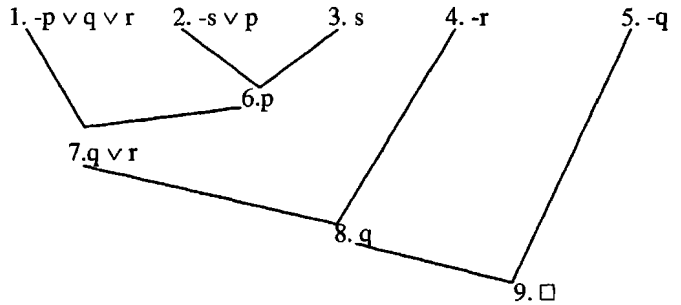
Rezolvenții 6, 7, 9 şi 10 aparțin lui S1, respectiv mulțimii clauzelor ce nu satisfac interpretarea I; rezolventul 8 aparține lui S2, respectiv clauzelor ce satisfac interpretarea I. Derivarea rezolutivă poate fi descrisă prin şirul 6(4,5), 7(6,3), 8(7,1), 9(2,5), 10(9,8), unde parantezele desemnează clauzele-părinți. Derivarea nu este liniară Rezolventul 9 nu are drept părinte pe 8, ci pe 2 şi pe 5.6, conform cerinței deosebirii de „grupe sangvine"

Aplicarea strategiei mulțimii de suport a condus la alt proces rezolutiv decât cel obținut anterior prin strategia explorării în lățime (vezi 16.6a).

c. Strategia priorității clauzelor unitare (the unit preference strategy)

Putem postula regula ca în alegerea perechilor de clauze să dăm prioritate clauzelor alcătuite dintr-un singur literal.

Fie $S = (-p \vee q \vee r) \wedge (-s \vee p) \wedge s \wedge p) \wedge s \wedge -r \wedge -q$.



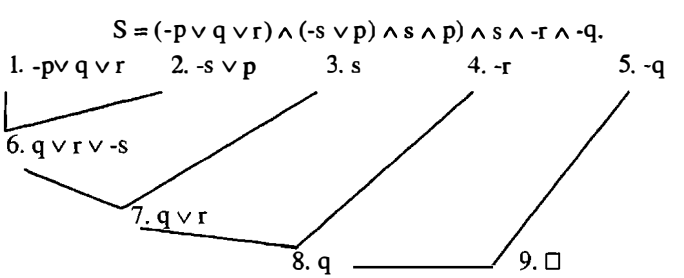
Exemplul de mai sus coincide cu exemplul 1 din 16.6 b.

d.Strategia utilizării complete a setului inițial de clauze (the liniar imput form strategy)

Rezolvitorii de probleme ştiu că într-o problemă fără date inutile, trebuie să se utilizeze toate datele sau informațiile problemei. Strategia utilizării complete a setului inițial de clauze reclamă captarea în procesul inferențial a tuturor clauzelor inițiale.

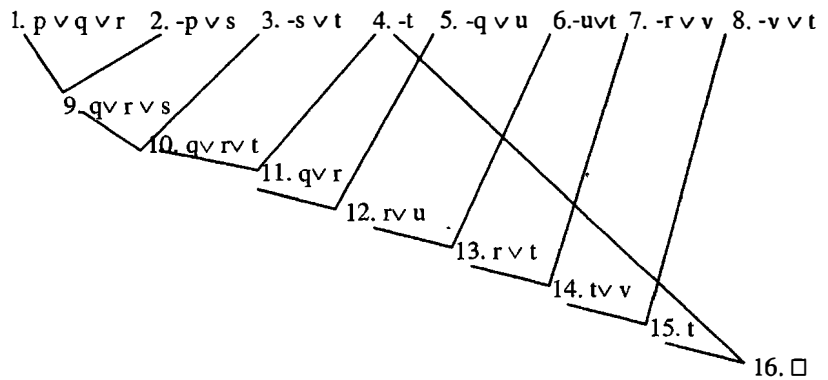
Reluăm ca o primă ilustrare pentru această strategie exemplul dat mai sus.

Exemplul 1.



Exemplul 2:

$S = (p \vee q \vee r) \vee (-p \vee s) \vee (-s \vee t) \wedge -t \wedge (-q \vee u) \wedge (-u \vee t) \wedge$
$\wedge (-r \vee v) \wedge (-v \vee t)$



Clauza 4 este utilizată de două ori: o dată pentru derivarea rezolventului 11 şi a doua oară pentru derivarea clauzei vide, formula 16.

Demn de reţinut este faptul că, atât în primul exemplu, cât şi în al doilea, s-a folosit şi o strategie de „străpungere", ultimul rezolvent servind de fiecare dată drept una dintre clauzele parentale pentru obţinerea rezolventului următor:

$$r_{i+1} = res_1(r_i, k_j) \text{ iar } k_j \in S.$$

Dacă convenim să utilizăm şi pentru analiza exemplului 2 de mai sus distincţia dintre rangurile sau nivelurile rezolvenţilor, ca în cazul strategiei explorării în lăţime (vezi 16.6a), unde am denumit prin R1 mulţimea rezolvenţilor imediaţi din S, prin R2 mulţimea rezolvenţilor obţinuţi din S1 = S∪ R1 şi aşa mai departe, atunci este util să observăm că rezolventul 9 aparţine lui R1, 10 lui R2, iar 15 aparţine lui R7 şi 16 lui R8.Pentru a se obţine clauza vidă s-au folosit, aşadar, rezolvenţi de rang mai avansat, între R1 şi R8.

Rangul sau nivelul rezolvenţilor ne indică amploarea străpungerii, măsura în care ne îndepărtăm de datele iniţiale ale problemei. În exemplele analizate mai sus , una dintre premise, k_j, este, aşadar, permanent din S; ea ţine legătura cu datele iniţiale, iar cealaltă este rezolventul ultim.

e. Strategia rezoluției semantice

Această strategie se întemeiază pe divizarea setului inițial de clauze S în două clase disjuncte, după modul în care acesta satisface sau nu un model sau o asignare de valori dată tuturor variabilelor propoziționale care apar în S.

Fie A = $\{p_1, p_2, \ldots, p_n\}$ lista variabilelor care apar în setul S și $v(p_1), v(p_2), \ldots, v(p_n)$ o asignare de valori de adevăr dată variabilelor din A. Atunci o interpretare I va putea fi descrisă ca o listă de literali proveniți din variabile din A, după următoarea regulă: $p_i \in I$, dacă $v(p_i) = 1$ și $-p_i \in I$, dacă $v(p_i) = 0$. Ca reguli de inferență, vom putea scrie:

$$\frac{p_i = 1}{p_i} \qquad \frac{p_i = 0}{-p_i}$$

Fiind dat un set de clauze disjunctive S (sau o formă normală conjunctivă), metoda strategiei semantice, aplicată în logica propozițiilor, poate fi rezumată prin următoarele reguli:

1. Identifică lista variabilelor care apar în setul de clauze disjunctive S și noteaz-o prin A_S ;

2. Construiește o interpretare arbitrară sau o atribuire de valori acordată variabilelor din S. Aceasta revine la a scrie lista variabilelor din A_S cu sau fără semnul negației;

3. Scindează setul de clauze S în două submulțimi de clauze:

$$S1 = \{k \in S: v(k) = 1\} \text{ și } v(k) = 1 \equiv \exists\, \ell(\ell \in I \cap k)$$

$$S2 = \{ k \in S. v(k) = 0\} \text{ și } v(k) = 0 \equiv \forall \ell(\ell \in k \supset \ell \notin I)$$

S1 desemnează clauzele ce satisfac interpretarea, iar S2 pe cele care nu o satisfac.

4. Alege o pereche de clauze, una din S1 și alta din S2, rezolubilă într-un literal ℓ și calculează-i rezolventul;

5. Verifică dacă rezolventul satisface interpretarea I și trece-l în setul S1, dacă o satisface; dimpotrivă, trece-l în setul S2, dacă acesta nu o satisface. Treci, din nou, la punctul 4.

Dacă setul de clauze S a provenit din negația unui raționament valid, atunci aplicarea strategiei rezoluție semantice la S va conduce cu certitudine la obținerea clauzei vide. Strategia rezoluției semantice este, în aceste cazuri, întotdeauna eficace.

Exemplul 1.

Fie $S^* = (p \vee q) \wedge (-p \vee r) \wedge -r \wedge (-q \vee s) \wedge (-s \vee r)$

1 2 3 4 5

un set de clauze provenit din negaţia unei formule ce formaliza un raţionament oarecare. Să se determine prin strategia rezoluţiei semantice dacă raţionamentul din care a provenit setul S este valid. Pentru aceasta procedăm astfel:

1. Identificăm lista variabilelor ce apar în setul S. A_S = {p, q, r, s}. Construim o interpretare arbitrară oarecare: v(p) = v(q) = v(r) = v(s) = 1, respectiv I = {p, q, r, s}.

2. Luând interpretarea I drept criteriu, scindăm setul S într-o clasă S1 ce satisface interpretarea I şi într-o clasă S2 ce nu o satisface:

$S^*1 = \{k_1, k_2, k_4, k_5\}$

$S^*2 = \{k_3\}$

Este uşor de observat că o clauză k satisface o interpretare I dacă aceasta conţine cel puţin un literal din I.

3. Aplicăm instrucţiunea 4 la seturile S*1 şi S*2. Calculăm rezolventul din clauzele k_3 şi k_2 şi obţinem $res_r(k_3,k_2) = k_6$. -p. Clauza k_6 nu satisface interpretarea I. În consecinţă, k_6 va fi trecută în S*2. Căutăm în S*1 o clauză ce conţine literalul p, opusul literalului clauză unitară k_6 .Găsim clauza k_1. Rezolventul obţinut din k_1 şi k_6 este k_7. q. Acesta satisface interpretarea I şi, deci, va fi inclus în setul S*1 şi trebuie să-i căutăm un literal opus, -q, în S*2. Dar S*2 nu conţine nici o clauză care să aibă ca element literalul -q. Această linie rezolutivă se încheie aici prin obţinerea rezolventului k_7, care nu are, pentru moment, continuare. Căutăm o altă pereche de clauze, care să satisfacă exigenţele rezoluţiei semantice. Găsim o nouă pereche de start din S: k_3 şi k_5 . Calculăm $res_r(k_3, k_5) = k_8$. -s. Adăugăm pe k_8 la setul S*2 al clauzelor ce nu satisfac interpretarea I şi căutăm în setul S*1 o clauză ce conţine literalul s. Găsim clauza k_4. Calculăm rezolventul din clauzele k_8 şi k_4 şi obţinem $res_s(k_8 ,k_4) = k_9$. -q. Încadrăm pe k_9 în S*2 şi căutăm în S*1 o clauză care să conţină literalul q. Găsim clauza k_7, căreia nu-i găsisem anterior o continuare. Calculăm $res_q(k_7,k_9) = k_{10}$. □

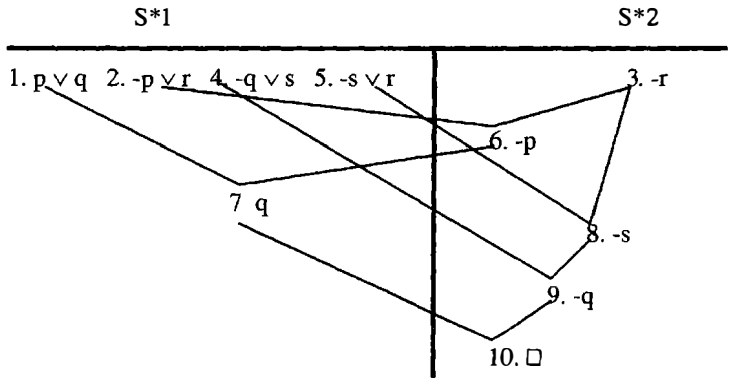
Demersul rezolutiv urmat în exemplul 1 poate fi reprezentat grafic în felul următor:

Fie $S^* = (p \vee q) \wedge (-p \vee r) \wedge -r \wedge (-q \vee s) \wedge (-s \vee r)$

1 2 3 4 5

$S^*1 = \{k_1, k_2, k_4, k_5\}$ I= {p, q, r, s}

$S^*2 = \{k_3\}$



Reprezentarea grafică a rezoluției semantice din exemplul 1

f. Strategii mixte

Strategiile rezolutive pot fi şi sunt adesea combinate unele cu altele în vederea potențării performanțelor rezolutive. Strategia explorării în lățime este opusă strategiei străpungerii" în care fiecare nou rezolvent este de alt rang. în consecință, cele două strategii nu pot fi combinate. Dar pot fi combinate, de exemplu, strategia mulțimii de suport cu cea a clauzelor unitare sau cu cea a utilizării complete a setului inițial de clauze sau cu cea a străpungerii în adâncime.

Metoda rezoluției este o specie de metodă deductivă şi nu doar un procedeu algebric de analiză a unei formule. Ea este un instrument de testare a validității unor raționamente, fiind aplicabilă prelucrării bazelor de cunoştințe. Ea este aplicabilă, deopotrivă, ca derivare rezolutivă progresivă, dar şi ca respingere rezolutivă, caz în care se porneşte de la premisele raționamentului sau de la datele problemei şi de la negația tezei de demonstrat. Dacă teza este demonstrabilă se obține clauza vidă. Dacă teza nu este demonstrabilă putem obține un model pentru formula ce descrie negația implicației dintre premise şi teza de demonstrat. Acestea sunt, de fapt, contraexemple sau cazuri de infirmare pentru formula supusă analizei. Dimpotrivă, dacă aplicăm procedeul respingerii rezolutive la o formulă ce nu e contradicție, i.e. la o contradicție din care am omis o clauză, nu vom obține clauza vidă, ci doar o clauză oarecare. Aceasta conjugată cu clauzele din S neutilizate şi supuse împreună operației negației ne vor da clauza omisă.

Strategiile rezolutive în logica propozițiilor au un rol minor. Ele devin importante în logica predicatelor de ordinul întâi. Pot juca,

evident, un rol şi în logica propoziţiilor, atunci când operăm cu un număr foarte mare de clauze disjunctive, când transpunem în logica propoziţiilor unele probleme de ordin tehnic.

7. Rezoluţia teleologică

Metodei rezoluţiei i s-a reproşat de către Alan Ramsey lipsa de intuitivitate şi o anumită îndepărtare de experienţa practică umană şi de limbile naturale. Spre deosebire, de exemplu, de calculul natural sau de metoda arborilor de decizie, metoda rezoluţiei, ca şi metoda Davis-Putnam, se întemeiază pe o anumită standardizare a expresiilor care face să dispară structura iniţială a raţionamentelor. Problema care ne-o punem este dacă putem să păstrăm caracterul standardizat al expresiilor şi să-i conferim, în acelaşi timp, metodei un grad sporit de raţionalitate. Noi credem că ne putem apropia de acest obiectiv prin ceea ce noi am numit anterior rezoluţie teleologică(vezi Cornel Popa, *Logica predicatelor*, Editura Hyperion, 1992, pp 149-157, 320-333).

O a doua cale de a apropia formele normale şi în particular clauzele Horn şi clauzele program de intuiţiile noastre este construirea unor arbori Şi-Sau şi a unor arbori de derivare pentru seturile de clauze program, respectiv pentru regulile Prolog. Asupra acestei căi de a redobândi intuitivitatea vom reveni într-un capitol următor.

Fie S un set de clauze şi A_S lista variabilelor care apar în S. Să presupunem că dorim să derivăm din S o clauză k în care apar variabilele A_k. Clauza k este clauza scop. Problema pe care ne-o punem este cum trebuie să procedăm pentru a obţine din setul S clauza k, printr-un număr mic de paşi şi, în acelaşi timp, procedura noastră să rămână clară, intuitivă?

Definim, mai întâi mulţimile de variabile(sau atomi) A_S şi A_k.

D1. $A_S = \{\ell \in L: \exists k(k \in S \wedge \ell \in k\}$

unde L este un alfabet al unei limbaj natural sau artificial. A_S este mulţimea atomilor distincţi care apar în setul de clauze.

D2. $A_k = \{\ell \in A_S: \ell \in k\}$.

A_K este mulţimea literalilor ce apar în clauza scop.

Pe baza acestora putem introduce două mulţimi relevante: mulţimea clauzelor de start şi mulţimea variabilelor(respectiv, literalilor) în care trebuie efectuată rezoluţia.

D3. $K_{START} = \{k \in S: \exists \ell(\ell \in A_k\}$

D4. $A_{RES} = A_S \setminus A_k$

Algoritmul rezoluţiei teleologice va consta din urmarea instrucţiunilor:

1. Determină mulțimea A_S a variabilelor care apar în setul inițial de clauze;
2. Determină mulțimea A_k a variabilelor care apar în clauza de demonstrat k;
3. Determină mulțimea literalilor de rezoluție, A_{RES};
4. Determină mulțimea clauzelor de start, K_{START};
5. Alege o clauză de start din K_{START}, k_{START} și dintre literalii acesteia un literal ℓ_1 opus literalului de start ce apare în k_{START};
6. Identifică între clauzele din S o clauză k_m în care apare literalul opusul literalului ℓ_1, respectiv $-\ell_1$;
7. Calculează $res_{\ell 1}(k_{START}, k_m)$;
8. Alege un literal de rezoluție ℓ_1 din $res_{\ell 1}(k_{START}, k_m)$ și treci apoi la instrucțiunea 6;
9. Dacă $res_{\ell 1}(k_{START}, k_m)$ nu mai conține nici un literal de rezoluție, atunci treci la instrucțiunea 5;
10. STOP cu SUCCES, dacă $res_{\ell 1}(k_{START}, k_m) = k$;
11. STOP cu EȘEC, dacă au fost epuizate toate clauzele de start și în cazul acestora toți literalii de rezoluție și $res_{\ell 1}(k_{START}, k_m) \neq k$.

Exemplul 1

$$S1 = \underset{1}{(p \vee -t)} \wedge \underset{2}{(s \vee t)} \wedge \underset{3}{(q \vee -s)} \wedge \underset{4}{(-p \vee w)} \wedge \underset{5}{(v \vee -w)} \wedge \underset{6}{(-r \vee -v)}$$

Dorim să derivăm din S clauza k1 = q ∨-r, i.e. $S \models_{res} q \vee -r$

Determinăm, conform instrucțiunii 1, mulțimea A_{S1} a variabilelor care apar în setul inițial de clauze:

$A_S = \{p, q, r, s, t, v, w\}$

La fel, calculăm, conform instrucțiunii 2, mulțimea A_{k1}

$A_{k1} = \{ q, -r\}$

Determinăm mulțimea literalilor de rezoluție, A_{RES} :

$A_{RES} = \{p, s, t, v, w\}$

Determinăm, potrivit instrucțiunii 4, mulțimea clauzelor de start, K_{START} :

$K_{START} = \{3, 6\}$, respectiv: $\{(q \vee -s), (-r \vee -v)\}$

Alegem, potrivit instrucțiunii 5, clauza 3 drept clauză de start și pe -s drept literal de rezoluție.

Căutăm, apoi, conform instrucțiunii 6, o clauză din S1 care să rezolve, în literalul -s, cu clauza de start 3 și găsim clauza 2.

Calculăm, conform instrucțiunii 7, res_{-s} (3, 6), i.e. res_{-s} (q ∨ -s, s ∨ t) şi obținem:

$$res_{s}(q \vee -s, s \vee t) = q \vee t.$$

Alegem, potrivit instrucțiunii 8, un literal de rezoluție din rezolventul obținut. Acesta trebuie să fie diferit de orice element din alfabetul de literali ai lui k. Îl alegem, deci, pe t.

Trecem apoi la instrucțiunea 6 şi căutăm perechea lui q ∨ t pentru o rezoluție în literalul t. Găsim clauza 1, ce conține pe -t. Calculăm, potrivit instrucțiunii 7,

$$res_{t}(q \vee t, p \vee -t) = p \vee q.$$

Trecem, din nou la instrucțiunea 6 şi căutăm în S (sau în clauzele rezolvent anterior derivate) o clauză care să rezolve cu p ∨ q în literalul p şi găsim clauza 4, respectiv clauza -p ∨ w.

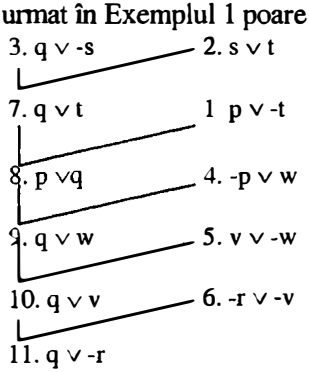
Conform instrucțiunii 7 calculăm pe res_{p}(p ∨ q, -p ∨ w) şi obținem ca nou rezolvent pe q∨ w.

Trecem, din nou, la instrucțiunea 6 şi căutăm în S o pereche pentru q∨ w care să rezolve cu aceasta în literalul w. Găsim clauza 5, respectiv pe v ∨ -w, care-l conține pe -w.

Mai departe, calculăm, potrivit instrucțiunii 7, res_{w} (q ∨ w, v ∨ -w) = q ∨ v. Potrivit aceloraşi instrucțiuni 6 şi 7, din noul rezolvent q ∨ v şi din clauza 6 , care este -r ∨ -v , printr-o rezoluție în literalul v, obținem rezolventul q ∨ -r.

Terminăm cu SUCCES conform instrucțiunii 10, căci rezolventul obținut, q ∨ -r, este identic cu clauza scop k1.

Demersul urmat în Exemplul 1 poare fi redat prin graful de mai jos:



Exercițiul 2

$$S2 = (p \vee v) \wedge (s \vee w) \wedge (-v \vee -w) \wedge (-q \vee r \vee -t) \wedge (-s \vee t)$$
$$\quad 1 \qquad 2 \qquad 3 \qquad 4 \qquad 5$$

Să se demonstreze prin rezoluție teleologică din S2 clauza: k 2 = p ∨ -q ∨ r

Primele patru instrucțiuni ne cer să definim trei mulțimi de literali și o submulțime de clauze din S.

A_{S2} = {p, q, r, s, t, v, w}

La fel, calculăm, conform instrucțiunii 2, mulțimea A_{k2}

A_{k2} = {p, -q, r}

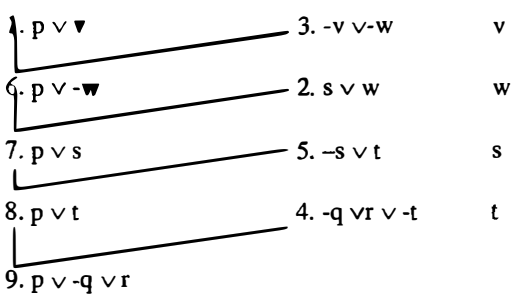
Determinăm mulțimea literalilor de rezoluție, A_{RES}

A_{RES} = { s, t, v, w}

Determinăm, potrivit instrucțiunii 4, mulțimea clauzelor de start, K_{START} :

K_{START} = {1, 4}, respectiv: { p ∨ v, -q ∨ r ∨ -t }

Alegem, dintre clauzele de start, conform instrucțiunii 5, clauza 1 și aplicăm apoi instrucțiunile 6-8 .Demersul rezolutiv poate fi ilustrat prin graful G2 redat mai jos.



Reprezentarea grafică a exemplului 2 sau graful G2

Exemplul 3

S3 = S2 = (p ∨ v) ∧ (s ∨ w) ∧ (-v ∨ -w) ∧ (-q ∨ r ∨ -t) ∧ (-s ∨ t)
1 2 3 4 5

Să se demonstreze, prin rezoluție teleologică, din setul de clauze S3 cauza k3 = -p ∨ s.

Întrucât S3 este echivalent cu S2 vom avea $A_{S3} = A_{S2}$:

A_{S3} = {p, q, r, s, t, v, w}

La fel, calculăm, conform instrucțiunii 2, mulțimea A_{k3} :

A_{k3} = {-p, s}

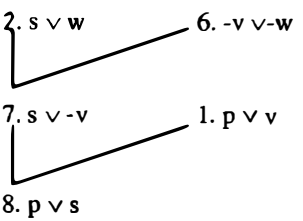
Determinăm mulțimea literalilor de rezoluție, A_{RES} :

$A_{RES} = \{q, r, t, v, w\}$

Determinăm, potrivit instrucțiunii 4, mulțimea clauzelor de start, K_{START} :

$K_{START} = \{2\}$, respectiv: $\{ s \vee w\}$

Apare o situație neașteptată: literalul -p, care apare în clauza de demonstrat k3 nu apare nici într-o clauză din S. Apare, în schimb, literalul s din k3 în clauza 2 din S. Dacă din clauza 2, ce conține pe s putem demonstra ca teoremă clauza unitară s, utilizând, desigur, strategia rezoluției teleologice, atunci, adăugând la principiul rezoluției o schemă logică de forma $A \supset (B \vee A)$, vom obține ca adevărată teza de demonstrat $-p \vee s$. Încercăm, așadar, să demonstrăm clauza unitară s. Pornim de la clauza 2.



Dar, pentru a-l obține din 8. pe s avem nevoie de o clauză unitară în -p sau cel puțin de o clauză ce-l conține pe -p și de alte clauze în literali opuși pentru a putea elimina literalii „de prisos" pentru ca, în cele din urmă, să rămână numai s. Dar p este un „literal pur" în S și, în consecință, nu este eliminabil.

Pentru a dovedi că din S nu se poate demonstra clauza $-p \vee s$, respectiv $-(S \vdash_{res} -p \vee s)$, este suficient să arătăm că există o interpretare I în care formula $S \wedge -(-p \vee s)$ devine adevărată. Construim astfel setul de clauze S3*, alcătuit din S3 și din negația tezei de demonstrat, $-p \vee s$:

$$S3^* = (p \vee v) \wedge (s \vee w) \wedge (-v \vee -w) \wedge (-q \vee r \vee -t) \wedge (-s \vee t) \wedge p \wedge -s$$

Dacă dăm lui S3* interpretarea $I = \{ p, -s, w, -v, -q\}$ sau $I = \{ p, -s, w, -v, r\}$, atunci acesta devine adevarat. Întrucât setul S3 poate coabita cu negația concluziei vom conchide că din S3 nu se poate demonstra $-p \vee s$. O interpretare ca cea de mai sus poate fi aflată prin metoda interpretărilor semantice succesive sau prin metoda

Davis-Putnam. Ea poate fi obținută și prin metoda rezoluției asumându-ne drept scop obținerea de clauze unitare, care considerate toate la un loc ne furnizează o interpretare în care setul inițial devine realizabil.

Observații

1. Primele 4 instrucțiuni din algoritmul prezentat mai sus sunt preliminare și definesc trei mulțimi de literali(alfabetul de variabile al setului de clauze S, alfabetul clauzei scop sau al tezei de demonstrat, mulțimea literalilor de rezoluție) și o submulțime privilegiată de clauze, submulțimea clauzelor de start.

2. Clauza 3. q $\vee$ -s din exemplul 1 a fost aleasă potrivit instrucțiunii 4 și definiției D3. În locul lui 3 puteam alege pe 6. -r $\vee$ -v, care conține pe -r și să luăm ca literal de rezoluție pe -v și în continuare să căutăm o clauză care să conțină pe v, cum este clauza 5.

3. Instrucțiunea 5 comportă două clase de alternative. Mai întâi o șansă de a alege este dată de posibilitatea existenței mai multor clauze de start și, în al doilea rând, de posibilitatea apariției într-o clauză a mai mult de doi literali. Oricum, literalul din clauza de start ce apare în clauza scop trebuie evitat a fi ales ca literal în care se va face rezoluția.

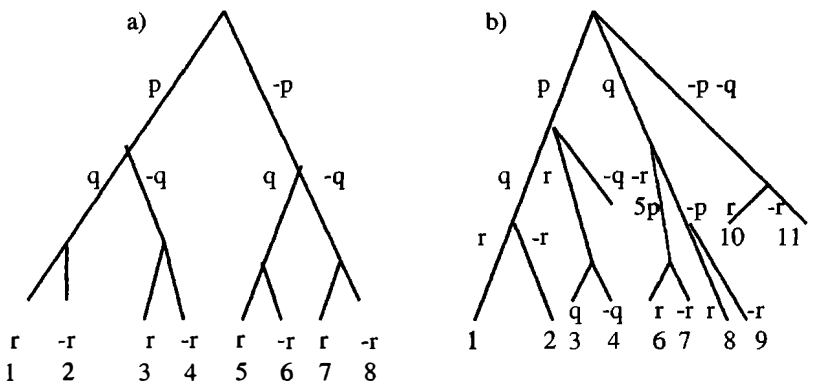
4. Instrucțiunea 6 poate comporta și ea mai multe alternative, căci putem avea în S sau în clauzele anterior demonstrate mai multe clauze ce conțin opusul literalului de rezoluție ales în pasul anterior.

5. Varianta de rezoluție teleologică prezentată în algoritmul propus (vezi instrucțiunea 8) este, totodată una de rezoluție liniară, căci clauza nou obținută este utilizată în pasul imediat următor ca una dintre clauzele „părinți" Evident, această restricție poate fi înlăturată, atunci când în clauza rezolvent nu se găsește nici un literal de rezoluție. Caracterul de rezoluție liniară este evidențiat de grafurile construite pentru exemplele 1 și 2. Puterea algoritmului propus nu se restrânge la soluționarea problemelor rezolubile prin rezoluție liniară (vezi în acest sens instrucțiunea 9).

6. Din analiza exemplului 3 se poate desprinde observația conform căreia dacă alfabetul unui set de clauze A_S nu conține toți literalii care apar în clauza scop sau clauza de demonstrat k, atunci aceasta nu poate derivata rezolutiv din S. În particular, în exemplul 3 nu a putut fi demonstrată din S3 clauza scop -p $\vee$ s, pe temeiul că S nu conținea nici o clauză în care să apară -p, p fiind literal pur în raport cu S.

8. Arborii închişi şi respingerile rezolutive

Interpretările semantice sunt frecvent redate prin arbori de interpretare. Pentru orice formulă având n variabile putem construi un arbore de interpretare sau semantic, cu ajutorul căruia putem explora toate interpretările ce pot fi date variabilelor sau atomilor formulei în cauză. Pentru o formulă f(p, q, r) vom avea drept arbori de interpretare arbori de forma:



Reprezentarea prin arbori a unei funcţii logice de trei variabile

Fiecare drum dintr-un arbore corespunde unei interpretări sau valorizări de atomi. Conjuncţia atomilor ce descriu drumul o vom numi *intensiune* a nodului terminal din drum.

Reprezentarea b) este uşor redundantă: 1 are aceeaşi „intensiune" ca 6 şi 3; 2 aceiaşi „intensiune" ca 7.

Un *arbore semantic complet* trebuie să satisfacă următoarele condiţii: 1) disjuncţia conotaţiilor sau intensiunilor asociate arcelor descendente dintr-un nod trebuie să exprime o tautologie; 2) fiecare drum trebuie să conţină toate variabilele sau atomii formulei; 3) etichetele asociate unui drum trebuie să nu conţină variabile propoziţionale sau predicate opuse.

Orice nod neterminal descrie o interpretare parţială desemnată de conjuncţia variabilelor (i.e atomilor) aflate deasupra nodului.

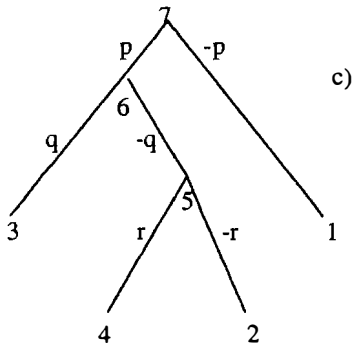
Fiind dat un set de clauze (incompatibile) S, vom spune că un nod oarecare este un *nod de infirmare al unei clauze k din S*, dacă şi numai dacă, intensiunea sa (dată de conjuncţia etichetelor de deasupra sa) implică negaţia clauzei k şi nici un nod anterior lui nu face aceasta.

Un *arbore semantic este închis* în raport cu un set de clauze S, dacă toate nodurile sale terminale sunt noduri de infirmare a unor clauze din S.

Exemplul 1

Fie $S = \{\overset{1}{p}, \overset{2}{q \vee r}, \overset{3}{-p \vee -q}, \overset{4}{-p \vee -r}\}$

c)



Nodul 1 este un nod de infirmare pentru clauza 1 din S. Într-adevăr, intensiunea lui 1, -p, este negația clauzei 1 din S. în mod analog, nodul 2 din graf cu intensiunea $p \wedge -q \wedge -r$, implică negația clauzei 2 din S; la fel nodul 4 din graf cu intensiunea $p \wedge -q \wedge r$, implică negația clauzei 4 $-p \vee -r$: $p \wedge -q \wedge r \rightarrow p \wedge r$, $p \wedge r \rightarrow -(-p \vee -r)$ (prin legile lui De Morgan); în sfârșit, intensiunea nodului 3 presupune negația clauzei 3 din S. Prin urmare, arborele c) de mai sus este un arbore închis în raport cu setul de clauze S din exemplul 1.

Numim *nod de inferență* sau *nod de rezoluție* un nod dintr-un arbore semantic din care pleacă două arce (sau muchii) terminate ambele cu noduri de infirmare. De exemplu, nodul 5 din arborele c) este un nod de inferență, căci ambele sale arce descendente, terminate în nodurile 4 și 2, sunt noduri de infirmare, clauzele 4 și 2 fiind rezolubile în literalul r.

Cum se construiește arborele închis asociat unui set de clauze incompatibile S?

Pentru a construi arborele semantic închis asociat unui set de clauze incompatibile S procedăm astfel:

1. Calculăm negația setului de clauze incompatibile S obținând FND, i.e. un set de clauze conjunctive, fiecare reprezentând negația uneia dintre clauzele disjunctive inițiale ale setului S.

2. Construim un model sau o interpretare pentru fiecare clauză conjunctivă obținută conform instrucțiunii 1, respectiv câte un nod de infirmare pentru clauza din S a cărei negație este clauza conjunctivă în cauză.

Începem operaţia cu clauzele unitare sau cu cele având cel mai mic număr de literali.

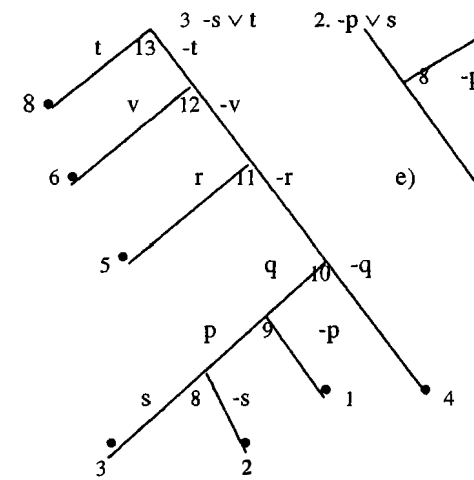
Marcăm fiecare nod de infirmare prin numărul de ordine al clauzei infirmate din S. De regulă, într-un arbore închis ultimele două noduri de infirmare definesc un nod de inferenţă, care, la rândul lui, împreună cu un alt nod de infirmare, defineşte un nou pas rezolutiv.

Exemplul 2:

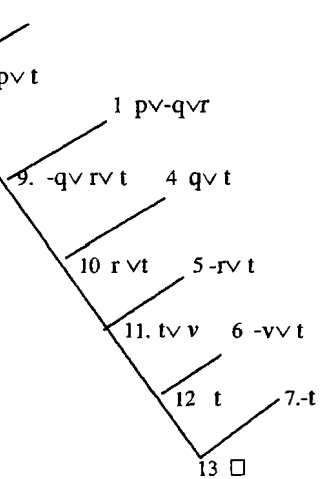
Fie $S = (p \vee -q \vee r) \wedge (-p \vee s) \wedge (-s \vee t) \wedge (q \vee t) \wedge (-r \vee v) \wedge (-v \vee t) \wedge -t$
1 2 3 4 5 6 7

$-S = -pq\text{-}r \vee p\text{-}s \vee s\text{-}t \vee -q\text{-}t \vee r\text{-}v \vee v\text{-}t \vee t$
1^0 2^0 3^0 4^0 5^0 6^0 7^0

d)



e)



Arborele închis asociat setului de clauze S din exemplul 2 (vezi d) se construieşte astfel: începem cu clauza unitară t care duce la un nod de infirmare marcat prin formula 7. Identificăm apoi în -S o clauză conjunctivă care conţine pe -t şi facem bifurcarea în v şi -v; se închide v-t ca negaţie a clauzei 6 şi continuăm bifurcarea în variabila r, r-v este negaţia lui 5 şi se continuă bifurcaţia în q şi -q ş.a.m.d.

Întrucât 3 şi 2 din (d) sunt noduri de infirmare având acelaşi punct de pornire, punctul lor de pornire 8 va fi un nod de inferenţă. La rândul lui, acesta va genera nodul de inferenţă 9; 9 împreună cu 4 va genera pe 10, iar 10 împreună cu 5 va genera pe 11, care, împreună cu 6, va

genera pe 12, care în sfârșit, împreună cu 7, va genera clauza vidă. Acest demers este reprodus de arborele de respingere rezolutivă (e).

Demn de reținut este faptul că arborele semantic închis ne furnizează algoritmul procesului rezolutiv asociat setului inițial de clauze.

9. Rezoluția duală, diviziunea și minimizarea

În afară de rezoluția clasică, definită de Robinson, ce utilizează clauze disjunctive și pornește, deci, de la formele normale conjunctive, există și o metodă rezolutivă duală, definită de Maslov, ce pornește de la formele normale disjunctive și operează cu clauze conjunctive, respectiv cu conjuncții de literali.

Ne propunem, în cele ce urmează, să introducem conceptele și principiile teoriei rezoluției duale. Astfel vom defini noțiunile de rezoluție duală și de rezolvent dual, vom demonstra teorema rezolventului dual, vom defini noțiunile de derivare rezolutivă duală și vom demonstra teorema derivării rezolutive duale.

Mai mult, vom introduce noi specii de rezoluție duală, între care rezoluția duală minimizatoare și rezoluția duală simetrică și vom conecta teoria rezoluției duale cu problema minimizării expresiilor logice și a circuitelor și schemelor logice și cu metodele de minimizare. O atenție aparte vom acorda racordării problematicii rezoluției duale simetrice cu teoria diviziunilor multinivelare și cu formele normale disjunctive perfecte.

9.1. Rezolvent dual și rezoluție duală. Specii de rezoluții duale.

Fie F o formă normală disjunctivă

$$F = F_1 \vee F_2 \vee \quad \vee F_n \tag{1.1}$$

unde $n \geq 2$ si F_i și F_j aparținând lui F sunt *clauze conjunctive*. Pe acestea le mai putem numi și *conjuncții elementare*. În logica propozițiilor, *literalii* pot fi variabile propoziționale cu sau fără semnul negației în față. În logica predicatelor, literalii pot fi atomi predicativi cu sau fără semnul negației în fața lor.

Definiția 1. Fiind date două clauze conjunctive F_i și F_j astfel că:

$$F_i = F_i' \wedge l_i \tag{1.2}$$

$$F_j = F_j' \wedge \neg l_j \tag{1.3}$$

și existând o substituție unificatoare σ ce satisface condiția:

$$l_i \sigma = l_j \sigma \tag{1.4}$$

numim *rezoluție dual*ă trecerea de la clauzele F_i și F_j la clauza

$$F' = F_i'\sigma \wedge F_j'\sigma \quad \text{sau} \quad rd(k_1,k_2) = k_1'\sigma \wedge k_2'\sigma \qquad (1\,5)$$

unde F' este *rezolventul dual* din clauzele F_i și F_j

Este ușor de observat că:

$$F_i' = F_i\,\sigma \setminus l_i\,\sigma \quad \text{sau} \quad k_1'\sigma = (k_1 \setminus \{l\})\sigma \qquad (1.6)$$

$$F_j' = F_j\,\sigma \setminus \sim l_j\,\sigma \quad \text{sau} \quad k_2'\sigma = (k_2 \setminus \{\sim l\})\sigma \qquad (1\,7)$$

Formulele de forma $C\sigma$ desemnează formulele factor, respectiv rezultatul aplicării substituției σ la formula inițială C. Prin σ am notat cea mai generală substituție unificatoare, respectiv o substituție care în raport cu orice altă substituție unificatoare τ ce unifică două sau mai multe expresii, satisface condiția $\sigma\lambda = \tau$, unde λ este tot o substituție.

Fiecare clauză conjunctivă are un număr determinat de literali. Convenim să numim *rang* al unei clauze conjunctive numărul ei de literali distincți. De exemplu, rn(pqr)=3. Dacă într-o clauză conjunctivă apar doi literali identici, aceştia pot fi contopiți prin principiul idempotenței $A \wedge A \equiv A$.

Introducem o relație de ordine a literalilor dintr-o conjuncție de literali concordantă cu sistemul de numerație binar. Primul literal va fi cel din dreapta şi va corespunde lui 2^0, respectiv numărului 1, cel de al doilea corespunde lui 2^1 şi, deci, numărului 2, cel de al treilea corespunde lui 2^2 şi deci numărului 4. Definim astfel poz $(l,X) = 2^X$ unde X = 0,1,2,3,...

Dăm un exemplu de rezoluție dual*ă* în logica predicatelor.

Exemplul 1:

$$F_i = P(a) \wedge R(z,x) \wedge Q(b)$$

$$F_j = S(z,y) \wedge \sim Q(z)$$

unde F_i şi F_j sunt clauze conjunctive dintr-o formă normal disjunctivă. Atunci:

$$F' = P(a) \wedge R(b,x) \wedge S(b,y)$$

este rezolventul dual din F_i şi F_j iar $\sigma = \{b/z\}$ este substituția unificatoare, iar perechea de literali în care s-a făcut rezoluția a fost Q(b) din F_i şi ~Q(z) din F_j. Pe post de F_i' a fost $P(a)\wedge R(z,x)$, iar pe post de F_j' a fost S(z,y).

Rezolventul dual poate fi şi el caracterizat similar rezolventului clasic cu ajutorul diferenței dintre mulțimea de literali. Convenind să punem k_1 în loc de Fi şi k_2 în loc de F_j şi să notăm tradițional literalul de rezoluție prin ℓ, putem defini rezolventul dual ca:

$$rd(k_1,k_2) = (k_1 \setminus \{\ell\})s \wedge (k_2 \setminus \{-\ell\})s \qquad (1.8)$$

unde substituția σ am notat-o de data aceasta prin s.

Definiția 2 a) Numim *rezoluție duală minimizatoare* o rezoluție duală în care rangul rezolventului este mai mic cu o unitate decât rangul maxim al clauzelor-părinți:

$$rn(rd(k_1,k_2)) = \max(rn(k_1),rn(k_2))-1 \qquad (1.9)$$

În (1.9) am notat rangul unei clauze prin rn.

b) Numim *rezoluție staționară* o rezoluție în care rangul clauzei rezolvent este egal cu rangul maxim al clauzelor părinți.

$$rn(rd(k_1,k_2)) = \max(rn(k_1),rn(k_2)) \qquad (1.10)$$

c) Numim *rezoluție amplificatoare* o rezoluție în care numărul literalilor distincți din clauza rezolvent este mai mare decât numărul maxim de literali al clauzelor-părinți.

$$rn(rd(k_1,k_2)) > \max(rn(k_1),rn(k_2)) \qquad (1.11)$$

d) Numim *rezoluție minimizatoare simetrică* o rezoluție minimizatoare în care:

$$rn((k_1 \setminus \ell)\sigma) = rn((k_2 \setminus -\ell)\sigma) \quad (\text{i.e.} rn(\ Fi'\sigma) = rn(Fj'\sigma)) \qquad (1.12)$$

Este ușor de observat că o rezoluție duală minimizatoare simetrică satisface condițiile (1.9) și (1.12). Rezoluția duală simetrică descrie o generalizare; ea face abstracție de un criteriu de evaluare descris printr-o pereche opusă de literali $<\ell,\sim\ell>$ reținând toate celelalte note comune.

Exemplul 2.1a:

k_1=p, k_2=¬p ∧q, k_1'=ϕ, k_2'=q, $rd(k_1,k_2)$=q, $rn(k_1)$=1, $rn(k_2)$=2

$rn(rd(k_1,k_2))$=1, $\max(rn(k_1),rn(k_2))$=2, $rn(rd(k_1,k_2))$=1

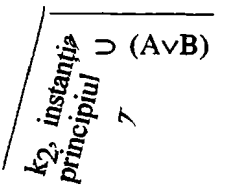
rd(p,~p∧q) este o rezoluție minimizatoare nesimetrică, pentru că rezolventul ei $rd(k_1,k_2)$ = q are rangul 1 și diferența dintre $\max(rn(k_1), rn(k_2))$ = 2 și $rn(rd(k_1,k_2))$ = 1 este 2 - 1 = 1. Este nesimetrică pentru că $rn(k_1') \neq rn(k_2')$, 0≠1.

Exemplul 2.2a

k_1=p∧q∧r, k_2=p∧~r∧s, $rn(k_1)$=3, $rn(k_2)$=3, $rd(k_1,k_2)$=p∧q∧s

$\max(rn(k_1),rn(k_2))$=3, $rn(rd(k_1,k_2))$=3

rd(p∧q∧r,p∧~r∧s)=p∧q∧s este o re logica propoziţiilor.



Exemplul 2.3a

k_1=pqr s, k_2=–stv, rn(k_1)=4, rn(k_2)=3,

rn(rd(k_1,k_2))=5, max(rn(k_1), rn(k_2))=

rn(rd(k_1,k_2))>max(rn(k_1), rn(k_2)), 5>

rd(pqrs,–stv)=pqrtv este o rezoluţie amplificat... rangul rezolventului ei este mai mare decât rangul oricăreia clauzele-părinţi.

Exemplul 2.4a

k_1=pqr, k_2=pq–r, rn(k_1)=3, rn(k_2)=3, rd(k_1,k_2)=pq,

max(rn(k_1),rn(k_2))=3, rn(k_1)=rn(k_2), rn(rd(k_1,k_2))=2, 2<3

rd(pqr,pq–r)=pq este o rezoluţie minimizatoare simetrică.

Exemplul 2.4b

k_1=p(x)∧q(y)∧t(z), k_2=p(a)∧q(u)∧~t(b), σ = {a/x,u/y,b/z}

k_1'σ =p(a)∧q(u), k_2'σ =p(a)∧q(u)

$rd_t(k_1,k_2)$=F'=p(a)∧q(u)∧p(a)∧q(u)=p(a)∧q(u) este o rezoluţie minimizatoare simetrică în logica predicatelor.

9.2. **Principiul rezoluţiei duale.** Principiul rezoluţiei duale va fi, spre deosebire de principiul rezoluţiei clasice, o inferenţă de la rezolvent la clauzele părinţi. O astfel de inferenţă o vom numi *inferenţă regresivă.*[*] Principiul afirmă că, dacă într-o interpretare I şi într-o atribuire de valori v va fi adevărată clauza rezolvent, instanţiată printr-o substituţie s, atunci cu certitudine, în aceeaşi interpretare şi într-o atribuire v', ce extinde pe v, va fi adevărată disjuncţia dintre clauzele părinţi k_1 şi

[*] Logicianul polonez A. Ligeza numeşte rezolventul dual drept rezolvent dual regresiv (*backward dual resolvent*). Vezi A. Ligeza, *Logical Foundations for Knowledge Based Control Systems. Knowledge, Representation, Reasoning and Theoretical Properties*, Raport LAAS, no 92269, Juillet 1992, Laboratoires d'Automatique et d'Analyse des Systèmes, Toulouse, p.29.

k_2, instanțiate prin aceeași substituție. Formularea de mai sus descrie principiul rezoluției duale sau teorema rezolventului dual.

Teorema 1 (Teorema rezolventului dual)

Dacă $\models v\ (k_1 \setminus \ell)s \wedge (k_2 \setminus \sim\ell)s$, atunci $\models^{v'} k_1 s \vee k_2 s$, (1.13)

unde pentru orice variabilă x_i ce apare în $(k_1 \setminus \ell)s$ sau în $(k_2 \setminus \sim\ell)s$ avem $v'(x_i)=v(x_i)$ și, în plus, în asignarea v' se fixează o valoare și pentru variabilele din literalul ℓ. Valorizarea v' este o extindere conservativă a valorizării v, respectiv ea păstrează toate valorizările date variabilelor individuale în v și în plus atribuie valori variabilelor individuale din ℓ.

Demonstrație:

Construim o demonstrație pentru principiul rezoluției duale exprimat în propoziția (1.13). Admitem adevărul rezolventului în interpretarea I și atribuirea v.

1) $\models v\ (k_1 \setminus \ell_i)s \wedge (k_2 \setminus \sim\ell_j)s$	*ipoteză*
2) $\models^{v}\ (k_1 \setminus \ell_i)s$	(E $\wedge$ 1)
3) $\models^{v}\ (k_2 \setminus \sim\ell_j)s$	(E $\wedge$ 1)

Potrivit lui 2), $(k_1 \setminus \ell_i)s$ este o conjuncție de atomi ce nu conține literalul ℓ_i, instanțiată prin substituția s, și care este adevărată în interpretarea I și asignarea v. Extindem conservativ asignarea v la asignarea v', valorizând în plus variabilele din literalul ℓ.

Reamintim că o *extindere conservativă* a unei valorizări v la o valorizare v' presupune păstrarea pentru toate variabilele x_i a asignărilor date de valorizare anterioară, în cazul nostru, v. Valorizarea v' va atribui, în plus, valori variabilelor din literalului ℓ_i (vezi C. Popa, *Logica Predicatelor*, Hyperion 1992).

Admitem că $\ell_i s$ este adevărat.

4) $v'(l_i s)=1$ ceea ce se mai poate scrie ca

4a) $\models^{v'} l_i s$

Atunci, din 2) și 4a) rezultă prin regula introducerii conjuncției

5) $\models^{v'} (k_1 \setminus \ell_i)s \wedge \models^{v'} \ell_i s$

De unde prin proprietățile diferenței dintre mulțimi și ale substituției, în particular prin ecuația $As \wedge Bs = (A \wedge B)s$ rezultă:

6) $\models v'\ k_1 s$,

ceea ce echivalează cu asertarea veridicității clauzei k_1, instanțiată prin substituția unificatoare s în valorizarea v'.

Dar, dacă este adevărată clauza k_1s, atunci prin $A \supset (A \vee B)$ vom avea:

7) $\models v' \; k_1s \vee k_2s$,

ceea ce înseamnă adevărul clauzelor părinţi.

Să admitem că avem:

8) $v'(\ell_i s) = 0$, respectiv $\models v' \; -\ell_i s$

Întrucât

9) $\ell_i s = \ell_j s$ vom avea:

10) $\models v' \; \ell_j s$

Cum în k_2 apare $\sim\ell_j s$, din 10) obţinem conform definiţiei semantice a negaţiei

11) $\models v' \sim\ell_j s$

Pe de altă parte, din 3), ţinând seama de faptul că v' este o extindere conservativă a asignării v, avem

12) $\models v' \; (k_2 \setminus \sim\ell_j)s$

Din 11) şi 12), pe baza regulii $I\wedge$ rezultă:

13) $\models v' \; (k_2 \setminus \sim\ell_j)s \wedge v' \sim\ell_j s$

De unde prin proprietăţile diferenţei dintre mulţimi şi ale substituţiei, în particular prin ecuaţia $As \wedge Bs = (A \wedge B)s$ rezultă:

14) $\models v' \; k_2s$

Mai departe, prin $B \supset (A \vee B)$ obţinem:

15) $\models v' \; k_1s \vee k_2s$

ceea ce concordă cu consecventul propoziţiei (1.13).

Cum şi în cazul că $\ell_i s=1$ şi în cazul că $\ell_i s=0$ obţinem din adevărul rezolventului adevărul clauzelor-părinţi (vezi 7) şi 15)) vom conchide că principiul rezoluţiei duale este o inferenţă regresivă, de la concluzie la premisele sau antecedenţii ei imediaţi.

Acest tip de demers inferenţial s-ar putea să-l găsim util în teoria argumentării, căci în acest fel de raţionament subiectul cunoscător sau argumentatorul porneşte de la teza de demonstrat către posibilele sale temeiuri sau raţiuni.

9.3. Derivare rezolutivă duală

Definiţia 3. Numim *derivare rezolutivă duală şi regresivă* a formulei F din mulţimea de formule F_{FND} o secvenţă de formule $F_1, ..., F_n$, astfel ca:

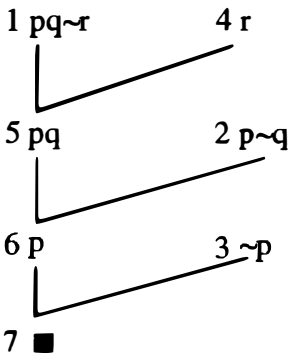
1) $F = Fn$

2) Pentru orice $j \in \{1,...,n\}$, f_j este sau un factor al formulei F_i sau un rezolvent dual regresiv din două formule F_{i1} şi F_{i2} unde i_1 şi i_2 preced pe j sau F_{i1} şi $F_{i2} \in F_{FND}$.

Reamintim că un factor al unei formule F, notat Fs, este rezultatul aplicării substituției s la formula F.

Dăm un exemplu de derivare rezolutivă duală în logica propozițiilor.

$F_{FND} = pq\sim r \vee p\sim q \vee \sim p \vee r$



Secvența ce exprimă derivarea rezolutivă este F1, F4, F5, F2, F6, F3, F7. F7 este clauza tautologică și este și clauza scop. Formula F5 este rezolventul dual din F1 și F4; F6 este rezolventul dual din F5 și F2; F7 este rezolventul dual din F6 și F3. F1, F4, F2 și F3 aparțin setului inițial de conjuncții elementare ce alcătuiesc forma normală disjunctivă a formulei sau mulțimii de formule pe care o testăm.

Într-o derivare rezolutivă duală rezolventul nou obținut se adaugă prin conectivul logic $\vee$ la disjuncția de conjuncții anterior existentă. Formula testată este tautologie, dacă obținem în final clauza tautologică, T.

Se observă că derivarea rezolutivă duală de mai sus satisface exigențele unei rezoluții liniare: fiecare rezolvent nou obținut este folosit imediat ca o clauză părinte în pasul rezolutiv imediat următor.

În sfârșit, se vede ușor că admiterea adevărului fiecărui rezolvent implică adevărul clauzelor sale părinte:

1. $T \supset (p \vee \sim p)$ $\quad (F7 \supset (F6 \vee F3))$
2. $p \supset (pq \vee p\sim q)$ $\quad (F6 \supset (F5 \vee F2))$
3. $pq \supset (pq\sim r \vee pqr)$ $\quad (F5 \supset (F1 \vee F4))$

Putem de asemenea conchide că:

4. $T \supset (pq\sim r \vee p\sim q \vee \sim p \vee r)$ $\quad (F7 \supset (F1 \vee F4 \vee F2 \vee F3))$

În final putem scrie că:

5. $F_{FND}1 \models_{RD} T$

Din forma normală disjunctivă $F_{FND}1$ se derivă rezolutiv dual regresiv clauza tautologică. Şi aceasta este perfect raţional, dacă spunem cititorului că $F_{FND}1$ este forma normală disjunctivă a axiomei lui Frege:

$(p \supset (q \supset r)) \supset ((p \supset q) \supset (p \supset r))$

Derivarea rezolutivă duală poate fi folosită, între altele, ca o metodă de decizie în logica propoziţiilor. Dacă formula testată este o tautologie logic-propoziţională, atunci din forma normal disjunctivă a formulei în cauză putem deriva dual regresiv clauza tautologică.

9.4. **Teorema derivării rezolutive duale.** Putem demonstra, ca şi în cazul rezoluţiei clasice, o teoremă a derivării rezolutive duale.

Teorema 2. Teorema derivării rezolutive duale. Fie F_{FND} o formulă în formă normal disjunctivă şi F o conjuncţie elementară derivată rezolutiv dual din F_{FND}. Atunci din F se deduce F_{FND} sau F_{FND} este o consecinţă logică sau o generalizare a lui F.

Demonstraţie:

Admitem că din F_{FND} derivă rezolutiv dual F, pe scurt $F_{FND} \models_{RD} F$. Atunci există o secvenţă de formule $F_1, F_2, \ldots, F_n$ astfel că $F \models_{RD} F_{FND}$ şi pentru orice $j \in [1,\ldots,n]$ F_j este un factor al unei formule F_i sau un rezolvent dual din două formule F_{i1} şi F_{i2} unde i, i1, i2 <j sau $F_i, F_{i1}, F_{i2} \in F_{FND}$.

(i) Să admitem că F_j este un factor al unei formule F_i anterioară. Atunci F_j se obţine din F_i printr-o substituţie. Deci, $F_j = F_i s$, unde s este o substituţie ce înlocuieşte variabilele libere din F_i prin termeni (constante individuale, variabile sau termeni compuşi). Putem, deci, spune că F_j derivă prin substituţie din F_i. F_i este mai general decât F_j.

(ii) Admitem că F_j este un rezolvent dual din două clauze conjunctive precedente k_1 şi k_2 printr-o rezoluţie într-o pereche de literali $\ell 1$ şi $\ell 2$ mediată de o substituţie unificatoare s. Atunci $\ell 1 s = \ell 2 s$ şi unul dintre literali este opusul celuilalt iar clauza rezolvent F_j va fi

$(k_1 \setminus \ell_{i1})s \wedge (k_2 \setminus \sim\ell_{i2})s$. Potrivit principiului rezoluției duale pentru orice interpretare I și pentru orice asignare de valori v ce face adevărat rezolventul $(k_1 \setminus \ell_{i1})s \wedge (k_2 \setminus \sim\ell_{i2})s$ există o extindere conservativă v' a lui v astfel că în interpretarea I și asignarea v' va fi adevărată disjuncția dintre k_1s și k_2s.

Dacă $\models_v (k_1 \setminus \ell_{i1})s \wedge (k_2 \setminus \sim\ell_{i2})s$, atunci $\models_{v'} k_1s \vee k_2s$ (1.7)

Pe scurt, veridicitatea rezolventului întemeiază predicatele clauzelor părinți sau a premiselor. Aceasta înseamnă că, din existența unei interpretări I și a unei asignării v ce întemeiază rezolventul terminal F_n, vom conchide asupra veridicității clauzelor-părinți din care a derivat aceasta, și anume F_{n-1} și F_{n-2}.

În măsura în care acestea sunt rezolvenți din alte clauze vom putea conchide asupra veridicității acestora, și tot așa mai departe până la forma normală disjunctivă inițială F_{FND}. Altfel spus, F_{FND} este o consecință logică a lui F, $F_{FND} \models_{RD} F$. Simbolul $\models_{RD}$ stă pentru relația de consecință rezolutivă duală. Consecința rezolutivă duală conduce între altele la sporirea gradului de generalitate a predicatelor derivate.

9.5. **O nouă interpretare a principiului rezoluției duale.** Pornim de la interpretarea admisă în mod curent în literatură pentru principiul rezoluției duale.

Pare nefiresc să interpretăm semantic principiul rezoluției duale ca o dependență a adevărului clauzelor părinți de adevărul clauzei rezolvent. Și în plus, ne pare la fel de nefiresc să conchidem de la o asignare mai restrânsă la o asignare mai extinsă.

Transpunem interpretarea admisă în literatură ca o schemă de inferență.

$$\frac{I \models_v (k_1 \setminus \ell)s \wedge (k_2 \setminus \sim\ell)s}{I \models_{v'} k_1s \vee k_2s\,,} \qquad (1.13)$$

Dacă rezolventul unei rezoluții duale este adevărat într-o interpretare I și într-o asignare de valori v, făcând uz de o substituție mgu s, atunci în aceiași înterpretare I și într-o extindere conservativă v' a lui v și făcând uz de aceeași substituție unificatoare s va fi

adevărată disjuncţia dintre clauzele părinţi k1 şi k2 înstanţiată prin aceiaşi substituţie s.

Principiul de mai sus poate fi rescris şi ca

$$\begin{array}{l} \models rd_l\,(k1, k2)\ v, s \\ \text{----------} \\ \models k1 \vee k2\ v', s \end{array} \qquad (1.14)$$

Principiul de mai sus poate fi transpus şi ca o implicaţie de forma: dacă antecedentul şi scriem ce-i deasupra barei orizontale, atunci şi scriem ce-i sub bară.

Dacă $I \models v\ (k_1 \setminus \ell)s \wedge (k_2 \setminus \sim \ell)s$, atunci $I \models v'\ k_1 s \vee k_2 s$, (1.13#)

şi respectiv ca:

Dacă $rd_l\,(k1, k2)\ v, s$, atunci $k1 \vee k2\ v', s$ (1.14.##)

Să aplicăm acum legea contrapoziţiei la aceste implicaţii.

Aceasta revine la a afirma că dacă este falsă disjuncţia dintre clauzele conjunctive k1 şi k2, într-o interpretare I şi o asignare de valori v' care este o extindere conservativă a asignării de valori v din rezolvent şi dată fiind o cea mai generală substituţie unificatoare s, atunci va fi fals în aceiaşi interpretare I şi în asignarea de valori v rezolventul dual din clauzele conjunctive k1 şi k2.

Ce rezultă din cele afirmate mai sus? Un lucru simplu şi important. *Rezoluţia dual conservă falsitatea de la clauzele părinţi la clauza rezolvent.*

Acest principiu poate fi extins de la principiul rezoluţiei duale la teorema rezoluţiei duale.

Rezoluţia dual este o schemă de inferenţă care conservă falsitatea de la clauzele părinţi la clauzele rezolvenţi.

Ea se întemeiază pe următoarele principii:

P1. *Dacă o f.n.d. este falsă, vor fi false toate conjuncţiile elementare sau modelele din care este alcătuită.*

$$A \vee B = 0$$

$$A = 0$$
$$B = 0$$

P2. *Dacă o conjuncţie elementară este falsă, atunci va fi fals cel puţin un termen al conjuncţiei, respectiv cel puţin un literal din conjuncţie.*

$A \wedge B = 0$

$A = 0 \vee B = 0$

P3. *Dacă un literal este fals, atunci dualul sau opusul său va fi adevărat.*

$A = 0$ $\quad -A = 0$

--- Legile negaţiei

$-A = 1$ $\quad A = 1$

P4. *Dacă n-1 dintre literalii unei conjuncţii elementare false sunt adevăraţi, atunci va fi fals cu certitudine ultimul literal din acea conjuncţie elementară*

$A1 \wedge A2 \wedge A3 = 0,\ A1 = A2 = 1$

$A3 = 0$

Aceste scheme de inferenţă ne permit să facem logic elementar şi intuitiv calcule de derivare rezolutivă duală.

În plus, aceste observaţii făcute mai sus ne permit să constituim alte variante de demonstrare a principiului rezoluţiei duale şi a teoremei derivării rezolutive duale.

Redăm mai întâi o demonstraţie alternativă a principiului rezoluţiei duale.

Convenim să scriem clauzele părinţi sub forma schematică

$k1 = A \wedge l$ (1)

$k2 = B \wedge -l$ (2)

unde A şi B sunt clauze conjunctive oarecare.

Pe baza exprimării schematice a clauzelor părinţi k1 şi k2 redate mai sus, rezolventul rezoluţiei duale va putea definit ca:

$rd_l(k1, k2) = rd_1(A \wedge l, B \wedge -l) = A \wedge B$

Noua formulare a principiului rezolventului, obţinută prin aplicarea legii contrapoziţiei la (1.13) va deveni:

Dacă $(A \wedge l \vee B \wedge -l) = 0$, atunci $A \wedge B = 0$ (4)

În această formulare, principiul rezoluţiei afirmă: Dacă clauzele-părinţi $(A \wedge l, B \wedge -l)$ sunt false, atunci rezolventul lor $A \vee B$ este fals.

Construim acum un text demonstrativ pentru acest principiu:

1. $(A \wedge l \vee B \wedge -l) = 0$	ip.
2. $A \wedge l = 0$	Def. $\alpha \vee \beta = 0$, 1
3. $B \wedge -l = 0$	Def. $\alpha \vee \beta = 0$, 1
4. $l = 1$	ip supl 1
5. $A \wedge 1 = 0$	RS 2, 4.
6. $A = 0$	P4, 5.
7. $A \wedge B = 0$	- $\wedge$, 6
8. $rd_l(k1, k2) = 0$	RE, 7.
9. $l = 0$	ip supl 2
10. $-l = 1$	Legea negaţiei
11. $B \wedge 1 = 0$	RE, 3, 10
12. $B = 0$	P4, 11
13. $A \wedge B = 0$	- $\wedge$, 12 qed

Dacă două clauze-părinţi (conjunctive şi rezolubile) sunt false, atunci este fals rezolventul lor dual. Rezoluţia dual conservă falsul de la clauzele părinţi la rezolventul lor.

Putem extinde acest rezultat de conservare a falsităţii de la principiul rezoluţiei duale la teorema derivării rezolutive duale.

Dacă S este o formă normală disjunctivă falsă sau S este un set de clauze conjunctive false şi din S derivă rezolutiv dual (Maslov) C, atunci C este fals. Cu alte cuvinte, derivarea rezolutivă dual conservă falsitatea formei normale disjunctive iniţiale în falsitatea consecinţelor duale obţinute.

Problema conservării falsităţii premiselor transpuse sub formă de clauze disjunctive în rezolvenţii duali obţinuţi poate fi redată schematic după cum urmează:

$$S = 0, \quad S \vdash_M C$$

$$C = 0$$

Demonstraţie.

1. $S = 0$ ip
2. $S \vdash_M C$ ip.
3. Dacă $k_i \in S$, atunci $k_I = 0$, conform regulii eliminării unei disjuncţii false (vezi mai sus regula P1), i.e. regula:

$A \vee B = 0$

---------- generalizată la disjuncţii cu n elemente, cu $2 \leq n \leq \infty$

$A = 0$

$B = 0$

4. Din $S \models_M C$ rezultă că există un șir de clauze disjunctive D = k1, k2, k3, ... kn, unde:

a) kn = C;

b) Orice ki din D, cu i între $1 \leq i \leq \infty$, $ki \in S$ sau

c) Dacă $ki \notin S$, atunci există kj și ks, cu $j < i$ și $s < i$ și există un literal *l* astfel că:

$$ki = rd_l(kj, ks)$$

5. Primele două elemente din șirul de clauze conjunctive din șirul D, k1 și k2 vor fi în mod necesar din D și potrivit principiului P1 menționat mai sus, vor fi false. Vom avea, deci:

5.1. k1 = 0

5.2. k2 = 0

6. Potrivit teoremei rezolventului demonstrată mai sus, dacă k1 = k2 = 0, rezolventul din ele, k3, va fi și el fals.

7. Potrivit principiului inducției, dacă o anumită proprietate este satisfăcută de primul element dintr-un șir P(1) și dacă, fiind satisfăcută de primele m elemente din șir, va fi în mod necesar satisfăcută și de m+1, P(m+1), atunci ea va fi satisfăcută de *toate* elementele din șir.

Prin urmare, condiția de falsitate va fi satisfăcută de orice element din șir, respectiv de orice derivare rezolutivă duală C.

Un exemplu. Să testăm prin derivare rezolutivă duală axioma lui Frege.

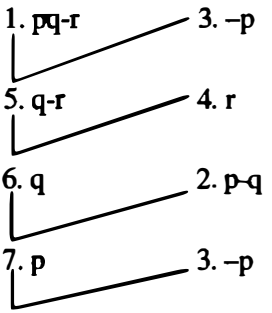
1. $S = (p \supset (q \supset r)) \supset ((p \supset q) \supset (p \supset r))$

Forma ei normală disjunctivă este:

1. $S' = pq\text{-}r \vee p\text{-}q \vee \text{-}p \vee r$
2. $S' = pq\text{-}r \vee p\text{-}q \vee \text{-}p \vee r = 0$

1 2 3 4

Construim o derivare rezolutivă duală pentru clauzele conjunctive din S':



Cum $\square = 0$ şi $\square = p \wedge -p$, rezultă că $p \wedge -p \supset 0$. De unde, prin regula negării termenilor unei identități putem scrie:

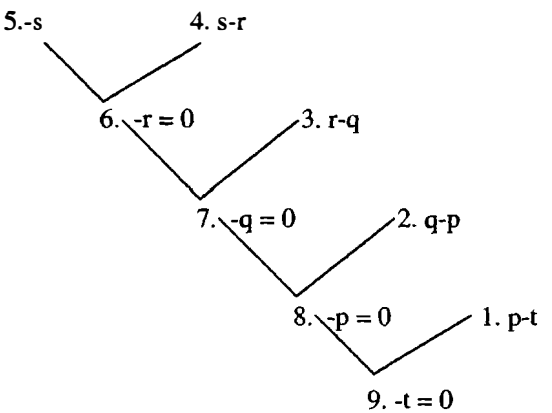
$$-(p \wedge -p) = -0$$
$$-p \vee p = 1.$$

Exemplul 2.

$$S = (-p \vee t) \wedge (-q \vee p) \wedge (-r \vee q) \wedge (-s \vee r) \wedge s = 1$$

$$S^* = \underset{1}{p\text{-}t} \vee \underset{2}{q\text{-}p} \vee \underset{3}{r\text{-}q} \vee \underset{4}{s\text{-}r} \vee \underset{5}{-s} = 0$$

Derivăm rezolutiv dual din S^* pe $-t = 0$ i.e. $S^* \models_M t$



Din 9, prin definiția negației avem din nou $t = 1$.

Este demn de reținut că rezoluția dual sau de tip Maslov, spre deosebire de cea clasică sau de tip Robinson, conservă valoarea de adevăr fals.

9.6. Formule reduse, clauze factor şi minimizări

Definiția 4. Fie C o conjuncție elementară $C = q_1 \wedge q_2 \wedge \ldots \wedge q_n$ unde q_i este un literal pentru $I = 1,\ldots,n$. Numim *formulă redusă* a formulei C şi o notăm prin |C| o formulă |C| ce satisface condițiile:

$$|C| \subseteq C \qquad (6.1)$$
$$|C| \equiv C \qquad (6.2)$$

și pentru orice literal $q \in |C|$

$$(|C| \setminus \{q\}) \cong C \qquad (6.3)$$

Definiția 4 este redată după A. Ligeza (p.25)

Formula redusă a unei conjuncții elementare se obține din formula inițială prin eliminarea unor literali și instanțierea altora printr-o substituție adecvată. Prin semnul $\cong$ am notat echivalența de semnificație sau echireferențialitatea.

Redusa unei formule conjunctive este o formulă instanțiată, semantic echivalentă cu formula inițială, având însă un număr mai mic de literali.

$F1 = r(a) \wedge q(a,b) \wedge r(x) \wedge q(z,y)$

$|F1| = r(a) \wedge q(a,b) \wedge q(b,c)$ $\qquad s1 = \{x/a, z/b, y/c\}$

$F2 = s(a) \wedge r(a,b) \wedge r(b,c) \wedge s(x) \wedge r(x,z)$

$|F2| = s(a) \wedge r(a,b) \wedge r(b,c)$ $\qquad s2 = \{x/a, z/b\}$

Se observă că redusa unei formule se obține prin substituții aplicate asupra formulelor deschise, astfel încât acestea să cadă peste (să fie unificate cu) alți literali instanțiați existenți în formula inițială și prin aplicarea legii idempotenței $A \wedge A \equiv A$, să se reducă complexitatea formulei.

Obținerea redusei este o operație de determinare și restrângere a unei conjuncții expandate la o conjuncție de literali instanțiați. Ea este diferită de aflarea unei formule factor cuprinzătoare, din care să poată fi obținuți prin substituție toți literalii formulei inițiale. De exemplu, clauza

$$F3 = p(x) \wedge p(f(y)) \wedge r(x,z)$$

poate fi redusă prin cel mai general unificator $s3 = \{x/f(y)\}$ la

$$F3' = p(f(y)) \wedge r(f(y),y)$$

care este clauza factor a lui F3.

După cum se vede, clauza factor F3' contopește, într-un singur literal, doi literali anteriori având același simbol predicativ. Atât redusa, cât și clauza factor sunt, într-un anumit sens, simplificări și contopiri de literali cu același simbol predicativ. Redusa face în plus o unificare cu un literal instanțiat existent în conjuncția inițială, în timp ce clauza factor contopește doi literali cu ajutorul unei substituții generice unificatoare.

Rezoluția duală conduce adesea la diminuarea numărului de literali din conjuncțiile elementare ale formei normal disjunctive inițiale în condițiile păstrării extensiunii expresiilor inițiale. Rezolutia

dual minimizatoare este o specie de rezoluție duală cu ajutorul căreia trecem de la conjuncții de rang n la conjuncții de rang n-1. Efectul acestor treceri repetate este minimizarea expresiilor.

Definiția 5. Spunem că facem o *derivare rezolutivă duală minimizatoare* dacă aplicăm la un set de clauze conjunctive inițiale în exclusivitate rezoluții duale minimizatoare.

Putem lesne construi un algoritm de minimizare a expresiilor logicii propoziționale și a logicii predicatelor, și, implicit, a schemelor cu contacte descrise de acestea, întemeiat pe metoda rezoluției duale. În logica propozițiilor, algoritmul va comporta doua etape principale:

1. Aplicarea principiului rezoluției duale minimizatoare la termenii formei normale disjunctive și calcularea tuturor rezolvenților duali minimizatori posibili: rezultatul acestei etape va consta în obținerea unor conjuncții elementare prescurtate sau minime.

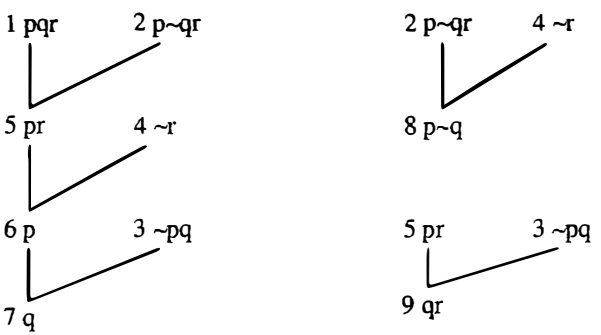
2. Construirea unui tablou care să verifice „acoperirea" de către conjuncțiile minimizate a tuturor conjuncțiilor elementare inițiale ce formează forma normală disjunctivă inițială.

Demn de reținut este faptul că procedeul de minimizare menționat nu pornește de la forma normală disjunctivă perfectă și de la analiza constituenților de 1, cum se procedează în mod curent în metoda diagramelor Karnaugh sau în metoda Quine-Clusky.

Exemplul 1

$F4 = pqr \vee p\neg qr \vee \neg pq \vee \neg r$
1 2 3 4

Derivăm din F3 rezolvenții duali:



Formula 4, ~r, este evident o formulă minimă. La fel sunt formule, minime 6 și 7. Derivarea din mulțimea de conjuncții elementare {pqr, p~qr, ~r} a literalului p este o derivare rezolutivă minimizatoare. La fel, derivarea din {p, ~pq} a lui q este tot o derivare rezolutivă minimizatoare. În schimb, derivarea din {pr, ~pq} a lui qr nu este o derivare rezolutivă minimizatoare. Tot derivare rezolutivă minimizatoare este și derivarea din {p~qr, ~r} a lui p~q. Setul de literali {p, q, ~r} și, respectiv, formula $p \vee q \vee \sim r$ este un candidat posibil pentru forma normală disjunctivă minimă F_{FND} a lui F4.

Verificăm acum veridicitatea acestei ipoteze prin tabloul de acoperire a termenilor formei normale disjunctive inițiale. Pentru aceasta vom plasa pe linii conjuncțiile minimizate și pe coloane conjuncțiile elementare inițiale.

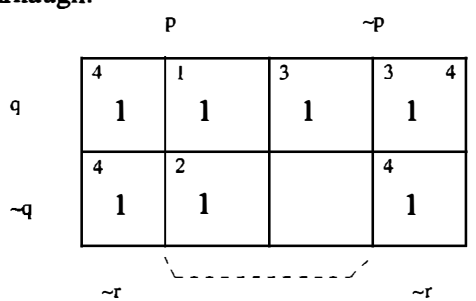
Conj. inițiale → Conj. reduse ↓	pqr	p~qr	~pq	~r
p	X	X		
q			X	
~r				X

Figura 1. Tabloul de acoperire pentru F4

Cei trei termeni minimizați acoperă pe toți cei patru termeni ai lui F4.

Formula $p \vee q \vee \sim r$ este expresia minimă a formulei F4.

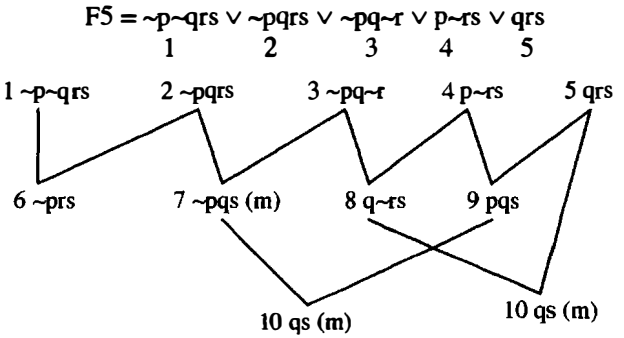
Putem, evident, verifica corectitudinea metodei noastre de minimizare prin rezoluție dual prin aplicarea la F4 a metodei diagramelor Karnaugh.



Este ușor de observat că formula $p \vee q \vee \sim r$ acoperă toate căsuțele marcate prin 1, prin care am descris inițial formula F4 (am

marcat în fiecare căsuţă şi conjuncţia elementară iniţială din care a provenit constituentul de 1).

Exemplul 2



Între termenii 1 şi 2 din F5 putem aplica o rezoluţie duală minimizatoare ce conduce la 6 ~prs. Între termenii 2 şi 3 are loc o rezoluţie duală minimizatoare nesimetrică încheiată cu 7 -pqs. Între 4 şi 5 are loc o rezoluţie comună încheiată cu rezolventul 9 pqs. Mai departe, termenii 7 şi 9, rezultaţi din rezoluţiile duale anterioare, permit o rezoluţie minimizatoare încheiată prin 10 qs. Este uşor de văzut ca 6 ~prs este o generalizare a lui 1 şi 2 şi că 1 şi 2 implică logic 6 (se face abstracţie de q). La fel, termenul 10 qs este o generalizare a termenilor 7 şi 9 (se face abstracţie de p). Dar 10 qs este şi o generalizare a lui 8 şi 5 (unde se face abstracţie de r). Putem, deci, conchide că 6 acoperă 1 şi 2, iar 10 acoperă pe 2 şi 5. Cu alte cuvinte, din 6 putem obţine prin expandare pe 1 şi 2, iar din 10 putem obţine prin expandare pe 2 şi 5. Expandarea sau adăugarea de noi literali complementari descrie o operaţie de *determinare,* similară diviziunii ce duce la identificarea speciilor complementare ale unui gen.

Conjuncţiile 3 şi 4 din F5 „neacoperite" de conjuncţiile minimizatoare 6 şi 10 şi având ambele rang 3 nu au o consecinţă comună de rang 2, deşi au o pereche complementară (p,~p) şi ~r apare şi în 3 şi în 4.

Construim tabelul de acoperire de către conjuncţiile elementare minimizate a termenilor disjuncţiei iniţiale F5.

Termeni init → Minitermeni ↓	-p~qrs	~pqrs	~pq~r	p~rs	qrs
6 ~prs	X	X			
10 qs		X			X

Figura 2. Tabloul de acoperire parţială pentru F5

Este evident ca ~prs ∨ qs nu acoperă integral pe F5. Pentru a construi un tablou în care miniterrnenii sau conjuncţiile reduse să acopere toţi termenii lui F5, vom adăuga între minitermeni termenii 3 şi 4 din F5, respectiv ~pq~r şi p~rs.

Astfel modificat, tabloul de acoperire devine:

Termeni init → Minitermeni ↓	-p~qrs	~pqrs	~pq~r	p~rs	qrs
6 ~prs	X	X			
10 qs		X			X
3 ~pq~r			X		
4 p~rs				X	

Figura 3. Tabloul de acoperire pentru F5

Expresia minimă este ~prs ∨ qs ∨ ~pq~r ∨ p~rs.

Aceeaşi expresie o vom găsi şi prin minimizarea lui F5 prin diagrame Karnaugh sau prin metoda lui Quine şi Clusky. Reproducem doar diagrama Karnaugh pentru F5.

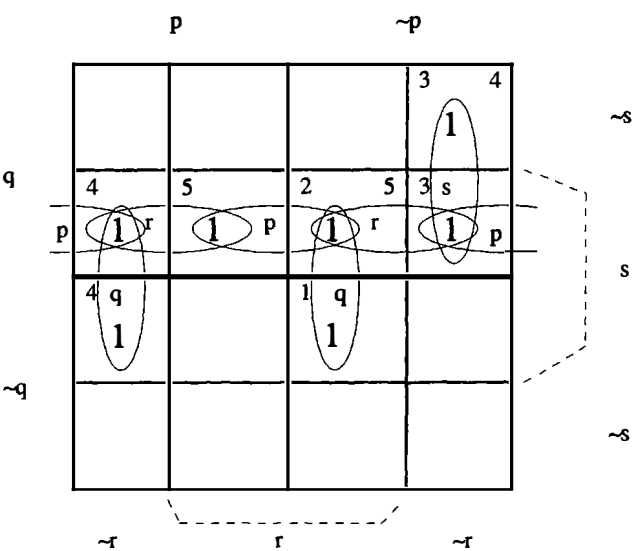


Diagrama Karnaugh pentru F5

Cifrele mici din colțurile careurilor cu 1 în centru desemnează conjuncția elementară din care a provenit constituentul de 1. Elipsele descriu simplificări. Am marcat în fiecare elipsă variabila simplificată sau de care s-a făcut abstracție. Cei patru de 1 scrişi pe linia a doua descriu pe 10 qs. Cele trei elipse verticale descriu, în ordine, de la stânga la dreapta, pe 4 p~rs, 6 ~prs şi 3 ~pq~r.

Se observă că qs, o conjuncție elementară de rangul 2, acoperă 4 careuri, respectiv intersecția dintre mulțimea careurilor asociate lui q şi lui s, în timp ce conjuncțiile elementare de rangul 3 (cu trei literali) acoperă fiecare câte două careuri. Conjuncțiile elementare de rangul 4 acoperă câte un singur careu. Un literal adevărat va fi descriptibil într-o diagramă cu 4 variabile prin 8 careuri acoperite cu 1. Clauza universală $\flat = p \vee \sim p$ va fi descriptibilă prin toate cele 16 careuri adevărate.

Orice rezoluție dual minimizatoare simetrică echivalează cu o relație de echireferențialitate între clauzele părinți şi clauza rezolvent. Din disjuncția clauzelor părinți derivă clauza rezolvent şi din clauza rezolvent se deduce disjuncția clauzelor părinți (din care s-a eliminat perechea opusă de literali şi asupra căreia s-a aplicat cea mai generală substituție unificatoare). Rezoluția dual simetrică descrie nemijlocit relații de minimizare de forma

$$AB \vee A\sim B \equiv A \wedge (B \vee \sim B) \equiv A \wedge 1 \equiv A \qquad (5.4)$$

9.7 Rezolvenți duali simetrici

Teorema 3. Teorema rezolventului dual şi simetric. Într-o rezoluție dual simetrică disjuncția clauzelor părinți şi rezolventul sunt echireferențiale:

$$sdr(k_1,k_2) \cong k_1 \vee k_2 \qquad (5.5)$$

Limităm pentru început demonstrația noastră la cazul logicii propoziționale.

Demonstrație

Fie $k_1 = k_1' \wedge \ell$ (a)

$k_2 = k_2' \wedge \sim\ell$ (b)

$k_1' = k_2'$ (c)

$$sdr(k_1,k_2) = k_1' \wedge k_2' = k_1' \wedge k_1' = k_1 = k_1' \wedge k_2' = k_2' \wedge k_2' = k_2 \quad (e)$$

$$k_1' = k_1' \wedge 1 = k_1' \wedge (\ell \vee \sim\ell) = k_1' \wedge \ell \vee k_1' \wedge \sim\ell = k_1 \vee k_2 \quad (f)$$

Este uşor de văzut că $sdr(k_1,k_2) = k_1' \wedge k_2'$ se obține pe baza principiului rezoluției duale aplicat la (a) şi (b). Mai departe, $k_1' \wedge k_1'$ se obține din $k_1' \wedge k_2'$ pe baza lui (c), iar k_1' se obține din $k_1' \wedge k_1'$

pe baza legii idempotenței. Deducerea din k_1' a lui $k_1 \vee k_2$ se face pe baza legilor algebrei logicii și a regulii substituirii echivalentelor mediată de egalitățile (a) și (b). Din (e) și (f) rezultă că din adevărul lui $sdr(k_1,k_2)$ rezultă adevărul lui $k_1 \vee k_2$ și invers.

Extindem acum demonstrația de mai sus la cazul logicii predicatelor. Condițiile inițiale vor fi:

$$k_1\sigma = (k_1' \wedge \ell_i)\sigma \qquad (a')$$
$$k_2\sigma = (k_2' \wedge \sim\ell_j)\sigma \qquad (b')$$
$$k_1'\sigma = k_2'\sigma \qquad (c')$$
$$\ell_i\sigma = \ell_j\sigma \qquad (d')$$
$$sdr(k_1,k_2) = k_1'\sigma \wedge k_2'\sigma = k_1'\sigma \wedge k_1'\sigma = k_1'\sigma$$
$$k_1'\sigma = k_1'\sigma \wedge (\ell_i\sigma \vee \sim\ell_j\sigma) = k_1'\sigma \wedge \ell_i\sigma \vee k_1'\sigma \wedge \sim\ell_j\sigma =$$
$$(k_1' \wedge \ell_i)\sigma \vee (k_1' \wedge \sim\ell_j)\sigma = (k_1' \wedge \ell_i)\sigma \vee (k_2' \wedge \sim\ell_j) = k_1\sigma \vee k_2\sigma$$

Între altele, s-a aplicat o lege logică ce descrie comportamentul substituției față de operația logică a conjuncției $A\sigma \wedge B\sigma = (A \wedge B)\sigma$. Pentru trecerea de la $k_1\sigma \vee k_2\sigma$ la $k_1 \vee k_2$ putem aplica substituția inversă σ^{-1}, ce restaurează clauza inițială. Cum, potrivit lui (c'), $k_1'\sigma$ este egal cu $k_2'\sigma$ putem arăta că din $k_2'\sigma$ derivă, de asemenea, $k_1\sigma \vee k_2\sigma$.

Relația de echireferențialitate desemnată prin $\equiv$ este o relație de echivalență. În consecință, ea va satisface proprietățile de reflexivitate, simetrie și tranzitivitate.

În cazul rezoluției duale simetrice, din fiecare dintre clauzele părinți se deduce clauza rezolvent:

$$k_1 \supset sdr(k_1,k_2) \qquad (5.6)$$
$$k_2 \supset sdr(k_1,k_2) \qquad (5.7)$$

și invers, din disjuncția clauzelor părinți se deduce clauza rezolvent:

$$k_1 \vee k_2 \supset sdr(k_1,k_2) \qquad (5.8)$$

Potrivit notațiilor de mai sus, (5.6) și (5.7) revin la:

$$k_1' \wedge \ell \supset k_1' \qquad (5.9)$$
$$k_2' \wedge -\ell \supset k_2' \qquad (5.10)$$

în logica propozițiilor și la:

$$(k_1' \wedge \ell_i)\sigma \supset k_1'\sigma \qquad (5.11)$$
$$(k_2' \wedge \sim\ell_j)\sigma \supset k_2'\sigma \qquad (5.12)$$

în logica predicatelor.

La rândul ei, 5.8 va fi exprimată în logica propoziţiilor prin:

$k_1' \wedge \ell \vee k_2' \wedge \sim\ell \supset k_1' \vee k_2$ (5.13)

şi cum $k_1' = k_2'$

$k_1' \wedge \ell \vee k_2' \wedge \sim\ell \supset k_1'$ (5.13a)

şi în logica predicatelor prin:

$(k_1' \wedge \ell_i)\sigma \vee (k_2' \wedge \sim\ell_j) \supset k_1'\sigma$ (5.14)

9.8. Rezoluţia dual simetrică şi minimizarea

Definim mai întâi noţiunea de derivare rezolutivă dual simetrică.

Definiţia 6. Fie F o formă normală disjunctivă perfectă. Spunem că din F *derivă dual rezolutiv simetric* clauza k si scriem F $I=_{sdr}$ k, dacă şi numai dacă, putem construi o secvenţă:

(1) $S = k_1, k_2, ..., k_n$ astfel că $k_n = k$

(2) Pentru orice k_i, $1 \leq i \leq n$, $k_i \in F$ sau daca $k_i \notin F$, atunci există două clauze k_j şi k_m cu j, m < I, astfel ca $k_i = sdr(k_j, k_m)$.

Orice aplicare a principiului rezoluţiei duale simetrice înseamnă reducerea a două clauze de rang n la o clauza de rang n-1. Este evident că o aplicare repetată a principiului menţionat mai sus înseamnă o reducere a numărului de termeni ai disjuncţiei iniţiale si a numărului de literali din conjuncţiile elementare derivate, în condiţiile conservării formulei iniţiale.

Algoritmul de minimizare pe care îl propunem primeşte la intrarea în sistem un set de conjuncţii elementare de diferite ranguri şi scoate la ieşirea din sistem un număr mai mic de termeni ai disjuncţiei având, totodată, un număr mai mic de literali în conjuncţiile elementare.

Etapele principale ale algoritmului de minimizare vor consta într-o serie de ordonări şi clasificări şi în aplicaţii ale principiului rezoluţiei duale simetrice.

1. Clasificăm conjuncţiile elementare după rangul lor. Conjuncţiile elementare de rang 1 sunt minitermeni ireductibili.

2. Clasificăm conjuncţiile din fiecare rang după numărul de atomi pozitivi pe care îi conţine. Astfel vom avea în fiecare rang conjuncţii ce conţin 0, 1, 2, 3, ... n literali pozitivi.

2a. Numim *clauze de conjuncţii învecinate* doua clase de conjuncţii de acelaşi rang astfel încât una conţine n literali pozitivi şi cealaltă n+1 literali pozitivi.

3. Se aplică principiul rezoluției duale simetrice la orice pereche de conjuncții învecinate de același rang ce satisface condiția $(k_1 - \{\ell\})\ \sigma = (k_2 - \{\sim\ell\})\ \sigma$ și unde ℓ și $-\ell$ ocupa aceeași poziție în conjuncțiile elementare părinți. Poziția literalului intr-o clauza se numără de la dreapta la stânga: în pqr, r va avea poziția 1, q poziția 2 s.a.m.d. Fiecărei poziții într-o clauză putem să-i asociem $b(y) = 2^X$ și X = 0, 1, 2, 4 .. constând dintr-o putere a lui 2.

Pentru un caz particular k_1 = pqr, k_2 = pq~r și $sdr(k_1, k_2)$ = pq avem:

$pqr \supset pq$	(eliminarea lui $\wedge$)
$pq\sim r \supset pq$	(eliminarea lui $\wedge$)
$pqr \vee pq\sim r \supset pq$	($pq\,(r \vee \sim r) = pqr \vee pq\sim r$)

Se remarcă, că între principiul rezoluției duale simetrice exprimat de proprietățile (5.6), (5.7) și (5.8) și formula (5.4) ce întemeiază operațiile logice de simplificare și expandare (introducerea unei noi variabile) există o deplină concordanță. Orice rezolvent simetric $sdr(k_1, k_2)$ reprezintă o simplificare sau reducere a doua clauze conjunctive inițiale la o clauza conjunctiva unică, având un rang mai mic cu o unitate decât clauzele ei părinți.

Exemplu

Fie F = pqr ∨ p~qr ∨ pq~r ∨ p~q~r ∨ ~qstv ∨ ~qst~v ∨ ~qs~tv ∨ ∨~qs~t~v

Împărțim termenii lui F in termeni de rang 3 și în termeni de rang 4:

F_{m3} = {pqr, p~qr, pq~r, p~q~r}

F_{m4} = {~qstv, ~qst~v, ~qs~tv, ~qs~t~v}

Partiționăm fiecare dintre clasele obținute F_{m3} și F_{m4} în clase determinate de numărul de literali pozitivi conținuți într-o conjuncție elementară.

F_{m3}

1	p~q~r	p~r, p~q	p
2	pq~r, p~qr	pq, pr	
3	pqr		

F_{rn4}

1	~qs~t~v	~qs~v, ~qs~t	~qs
2	~qst~v, ~qs~tv	~qst, ~qsv	
3	~qstv		

Degajăm din perechile de clauze învecinate din acelaşi rang rezolvenţi duali simetrici şi îi scriem în coloana imediat următoare.

9.9. **Dualitatea rezoluţiilor Robinson şi Maslov**

Rezoluţia Robinson este o metodă de prelucrare a bazelor de cunoştinţe transpuse în forme normale conjunctive sau seturi de clauze disjunctive. Rezoluţia Maslov este o metodă de prelucrare a bazelor de cunoştinţe transpuse în forme normale disjunctive, respectiv în disjuncţii de literali conectaţi între ei prin conjuncţie. Aceste conjuncţii de literali pot fi redate şi ca liste de literali ce satisfac mulţimea iniţială de formule (i.e. „modele" sau situaţii) ce descrie baza de cunoştinţe relaţională .

Bazele de cunoştinţe descriu date factuale şi relaţii de dependenţă dintr-un anumit domeniu din realitate. Una şi aceiaşi bază de cunoştinţe poate fi prelucrată prin ambele metode rezolutive, rezultatele obţinute fiind concordante.

Rezoluţia Robinson porneşte de la formele normale conjunctive sau de la seturi de clauze disjunctive. Rezoluţia Maslov sau rezoluţia duală porneşte de la formele normale disjunctive sau de la seturi de clauze disjunctive.

Rezoluţia Robinson conservă adevărul de la premise sau date la derivările rezolutive. Dimpotrivă, rezoluţia Maslov conservă falsitatea de la premise la consecinţe.

Ambele metode sunt metode analitice ce introduc standardizări şi pregătesc demonstraţia automatizată.

Ambele sunt scheme de inferenţă generalizate şi maşini inferenţiale performante ce pot fi nuanţate prin numeroase strategii rezolutive.

Cele două principii rezolutive sunt structural asemănătoare şi se poate trece de la unul la celălalt prin reguli de dualitate.

Acest lucru se poate vedea lesne dacă alăturăm cele două principii rezolutive:

Principiul rezoluţiei Robinson	*Principiul rezoluţiei Maslov*
$(l \vee A) \wedge (\neg l \vee B)$ f.n.c	$(l \wedge A) \vee (\neg l \wedge B)$ f.n.d.
$A \vee B$	$A \wedge B$
Conservă adevărul descendent	Conservă falsul descendent
$S = 1,\ S \vdash_R D$	$S = 0,\ S \vdash_M D$
$D = 1$	$D = 0$
	Conservă adevărul ascendent

Cele două metode rezolutive sunt legate între ele printr-un raport de dualitate şi se poate trece de la o metodă la alta cu ajutorul legilor sau regulilor de dualitate. Redăm aceste reguli sub forma unor scheme de inferenţă. Convenim asupra notaţiilor: α, β, γ sunt formule bine formate în logica propoziţiilor, T desemnează tautologia, C, contradicţia, α^* duala lui α, semnul $\vdash \alpha$ asertarea lui α

$\vdash \alpha = T$	$\vdash \alpha = T$	$\vdash \alpha = 1$	$\vdash ((\alpha)^*)^*$
R1 ------	R2. ----------	R3.----------	R4 ----------
$\vdash \alpha^* = C$	$\vdash \neg(\alpha^*) = T$	$\vdash \neg(\alpha^*) = 1$	$\vdash \alpha$
$\vdash \alpha = 0$	$\vdash \alpha \supset \beta = T$	$\vdash \alpha \equiv \beta = T$	$\vdash \alpha^* \equiv \gamma, \vdash \beta^* \equiv \gamma$
R5 -------	R6. ----------	R7. ----------	R8. ----------
$\vdash \alpha^* = 1$	$\vdash \beta^* \supset \alpha^* = T$	$\vdash \alpha^* \equiv \beta^* = T$	$\vdash \alpha \equiv \beta$

Tabloul 1. Regulile dualităţii

În limba română semnificaţia acestor reguli este:

R1:-Dacă α este o tautologie, atunci duala sa este o contradicţie.

R2. Dacă o formulă α este tautologie, atunci şi negaţia dualei ei este tautologie .

R3. Dacă o formulă α este adevărată, atunci şi negaţia dualei sale este adevărată.

R4. Din duala dualei unei formule se poate conchide asupra formulei iniţiale.

R5. Dacă α este falsă atunci duala sa α^* este adevărată.

R6. Dacă $\alpha \supset \beta$ este o tautologie, atunci contrapoziţia dualelor termenilor implicaţiei, $\beta^* \supset \alpha^*$, va fi o tautologie.

R7. Dacă o echivalență este tautologie, $\alpha \equiv \beta = T$, atunci și echivalența dintre dualele termenilor echivalenței, $\alpha^* \equiv \beta^*$, este tautologie.

R8. Dacă dualele a două formule α și β, adică α^* și β^* sunt echivalente cu o a treia formulă γ, atunci α este echivalentă cu β.

Adăugăm câteva reguli de dualitate ale derivărilor rezolutive Robinson și Maslov.

$\vdash \alpha = T$	$\vdash \alpha = T$	$\vdash \alpha_{NC} = T$	$S \vdash_R D$
R9 ----------	R10 ----------	R11 ----------	R12 ----------
$\alpha_{ND} \vdash_M T$	$\alpha_{NC} \vdash_R T$	$\vdash \alpha_{ND} = T$	$-S^* \vdash_M -D^*$

Tabloul 2. Dualitatea în derivările rezolutive

R9. Dacă α este o tautologie, atunci din forma sa normală disjunctivă derivă rezolutiv dual (Maslov) o tautologie ($p \vee -p$).

R10. Dacă din S derivă rezolutiv Robinson D, atunci din negația dualei lui S derivă resolutiv Maslov negația dualei lui D.

Pot fi, desigur, formulate și alte relații de dependență de forma:

R11. Dacă α este o tautologie, atunci din forma sa normală conjunctivă derivă rezolutiv (Robinson) o tautologie ($p \vee -p$).

R12. Dacă forma normală conjunctivă a unei formule α este o tautologie (T), atunci forma sa normală disjunctivă este tot o tautologie.

Datele unei probleme sau bazele de cunoștințe pot fi prezentate deopotrivă în forme normale disjunctive sau în forme normale conjunctive.

În funcție de forma de prezentare noi putem face uz de metoda rezoluției clasice sau de tip Robinson sau putem face uz de baze de cunoștințe duale.

Bazele de cunoștințe descriu deopotrivă relații de dependență dintr-un anumit domeniu din realitate. Reproducem câteva astfel de raporturi de dependență dintr-o carte despre *Sisteme expert de gestiune* (Editura Romcart 1993), semnată de Dorin Zaharie și Pavel Năstase.

1. Dacă importul crește și exportul scade, atunci balanța externă nu este echilibrată.

2. Dacă puterea de cumpărare crește sau producția este constantă sau producția scade, atunci prețul crește și importul crește.

3. Dacă prețul crește și productivitatea este constantă, atunci exportul scade.

4. Dacă productivitatea este constantă sau productivitatea scade și sarcinile cresc, atunci prețul crește și investițiile scad.

5. Dacă investițiile scad, atunci productivitatea este constantă.

6. Dacă preţul creşte şi producţia este constantă, atunci exportul scade.

7. Dacă timpul de lucru scade, atunci producţia scade.

8. Dacă producţia scade sau producţia este constantă şi sarcinile cresc, atunci investiţiile scad.

Formalizăm aceste raporturi de dependenţă prin implicaţii în logica propoziţiilor. Admitem pentru aceasta convenţiile:

importul creşte= p;
exportul scade= q;
balanţa externă este dezechilibrată (i.e. balanţa externă nu este echilibrată) = r
puterea de cumpărare creşte = s;
producţia este constantă = t;
producţia scade = v;
preţul creşte = u
productivitatea este constantă = p1;
productivitatea scade = q1;
sarcinile cresc = r1;
investiţiile scad = s1
timpul de lucru scade = t1;

1. $(p \wedge q) \supset r$
2. $(s \vee t \vee v) \supset (u \wedge p)$
3. $(u \wedge p1) \supset q$
4. $((p1 \vee q1) \wedge r1) \supset (u \wedge s1)$
5. $s1 \supset p1$
6. $(u \wedge t) \supset q$
7. $t1 \supset v$
8. $((v \vee t) \wedge r1) \supset s1$

Transpunem formulele de mai sus sub formă de clauze disjunctive sau clauze Robinson.

1. $-p \vee -q \vee r$
2.1. $-s \vee u$
2.2. $-s \vee p$
2.3. $-t \vee u$
2.4. $-t \vee p$
2.5. $-v \vee u$
2.6. $-v \vee p$
3. $-u \vee -p1 \vee q$
4.1. $-p1 \vee -r1 \vee u$
4.2. $-p1 \vee -r1 \vee s1$

4.3. ¬q1 ∨ -r1 ∨ u
4.4. –q1 ∨ -r1 ∨ s1
5. –s1 ∨ p1
6. –u ∨ -t ∨ q
7. –t1 ∨ v
8.1. –r1 ∨ -v ∨ s1
8.2. –r1 ∨ -t ∨ s1

Toate clauzele de mai sus sunt clauze Horn.

Clauzele Horn pot fi redate uşor ca instrucţiuni Prolog. Programul cu aceste clauze a fost trecut în calculator cu numele Zaharia 3.

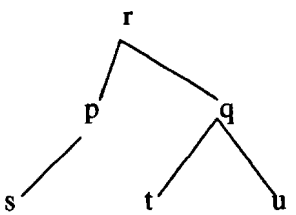
```
s.
t.
u.
r:- p, q.
p:- v.
p:- s.
p:- t.
q:- u, t.
q:- u, p1.
u:- s.
u:- t.
u:- v.
u:- p1, r1.
u:- q1, r1.
p1:- s1.
s1:- p1, r1.
s1:- r1, v.
s1:- r1, t.
s1:- q1, r1.
v:-t1.
```

Construim setul de clauze disjunctive:

$$S = \underset{1}{(-p \vee -q \vee r)} \wedge \underset{2}{(-s \vee u)} \wedge \underset{3}{(-s \vee p)} \wedge \underset{4}{(-t \vee u)} \wedge \underset{5}{(-t \vee p)} \wedge \underset{6}{(-v \vee u)} \wedge$$
$$\wedge \underset{7}{(-v \vee p)} \wedge \underset{8}{(-u \vee -p1 \vee q)} \wedge \underset{9}{(-p1 \vee -r1 \vee u)} \wedge \underset{10}{(-p1 \vee -r1 \vee s1)} \wedge$$
$$\wedge \underset{11}{(-q1 \vee -r1 \vee u)} \wedge \underset{12}{(-q1 \vee -r1 \vee s1)} \wedge \underset{13}{(-s1 \vee p1)} \wedge \underset{14}{(-u \vee -t \vee q)} \wedge$$
$$\wedge \underset{15}{(-t1 \vee v)} \wedge \underset{16}{(-r1 \vee -v \vee s1)} \wedge \underset{17}{(-r1 \vee -t \vee s1)}$$

Clauzele s, t, u de la începutul programului nu sunt dintre cele stipulate de Zaharie şi Năstase, ci sunt adăugate de noi ca modele sau intrări în sistem. Ele ne-au fost sugerate de construirea arborelui de derivare care să poată demonstra concluzia r, după care balanţa de plăţi este dezechilibrată.

Argumentarea tezei după care balanţa externă este dezechilibrată ar putea arăta astfel:



În limba naturală vom spune că balanţa externă este dezechilibrată, deoarece importul creşte şi exportul scade. Mai departe, importul creşte deoarece puterea de cumpărare creşte (şi aceasta a crescut nemotivat economic, prin mărirea lefurilor), iar exportul a scăzut deoarece producţia a rămas constantă şi preţul produselor propuse pentru export a crescut.

Arborele de derivare al unui set de clauze Horn este relevant pentru linia de demonstrare pe care o urmează un program Prolog.

Arborele reprodus mai sus este un subarbore al celui ce poate fi construit pentru cele 17 clauze enumerate care descriu „legile interne" ale sistemului la care trebuie, evident, adăugate „intrările „ sau situaţiile acţionale sau „modelele" (liste de literali de pe frunze) care implică adevărul tezei de demonstrat. Arborele de mai sus conţine trei clauze generice: 1, 2.2, 6 şi clauzele factuale adăugate de noi ca valori de intrare: s, t, u. Din setul de clauze disjunctive S, clauzele utilizate pentru întemeierea lui r au fost clauzele 1, 3, 14, la care sau adăugat literali de pe frunzele arborelui de întemeiere,s, t şi u.

Forma normală disjunctivă care descrie argumentarea de mai sus este S1:

$$S1 = \underset{1}{(-p \vee -q \vee r)} \wedge \underset{3}{(-s \vee p)} \wedge \underset{14}{(-u \vee -t \vee q)} \wedge \underset{18}{s} \wedge \underset{19}{t} \wedge \underset{20}{u} = 1$$

unde primele trei clauze sunt reguli şi ultimele trei sunt ipoteze iniţiale sau fapte.

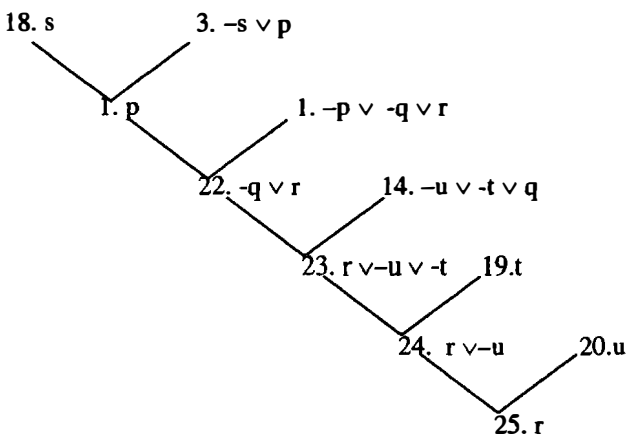
Din S1 putem deriva prin rezoluţii Robinson r , precurtat $S1 \vdash_R r$

Arborele de mai sus trebuie citit ca un arbore SI-SAU. Arborii ŞI-SAU descriu, între altele, forma normală disjunctivă sau mulţimea ipotezelor factuale care duc la demonstrarea tezei din rădăcină. Arborele de derivare ŞI-SAU redă deopotrivă argumentarea tezei şi demonstrarea Top Down a tezei din rădăcină.

Arborii de derivare ne arată cum putem construi probleme de logica propoziţiilor şi logica predicatelor.

Cercetăm, în continuare, pe baza setului S1 relaţia dintre derivarea rezolutivă de tip Robinson şi relaţia de derivare rezolutivă duală sau de tip Maslov, pe care o vom nota $S1 \models_M r$.

Derivăm în sens Robinson din S1 pe r.



Derivarea de mai sus este o derivare rezolutivă liniară şi este, evident, de tip Robinson care a pornit de la un set de clauze adevărate şi se încheie tot cu o clauză adevărată, r.

Construim duala lui lui S1, S1*. Aceasta înseamnă înlocuirea în S a literalilor prin opuşii lor, a conectivelor booleene prin dualele lor şi a valorilor de adevăr binare prin opusele lor. Convenim să nu mai scriem semnul conjuncţiei ∧ şi să scriem literalii alăturaţi, ca în produsele din algebră.

$S1 = (-p \vee -q \vee r) \wedge (-s \vee p) \wedge (-u \vee -t \vee q) \wedge s \wedge t \wedge u = 1$

$S1^* = pq\text{-}r \vee -ps \vee -qtu \vee -s \vee -t \vee -u = 0$

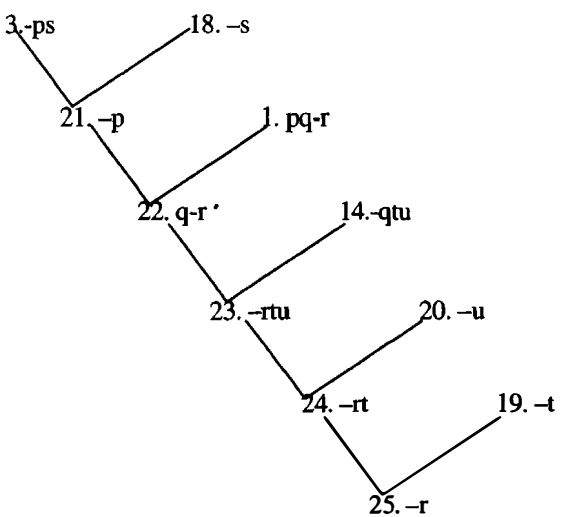
1 3 14 18 19 20

Dacă $S1^* = 0$, cum $S1^*$ este o disjuncţie, fiecare dintre termenii ei vor fi 0. Prin urmare, clauzele unitare 18, 19, 20 vor fi fiecare în

parte false şi, prin urmare, vor fi adevărate s, t şi u. Cum 14 este şi ea falsă, căci e falsă S1*, si t şi u sunt adevărate, falsitatea lui 14 se datorează falsităţii lui –q. Prin urmare, q este adevărată. La fel, cum 3 este falsă şi s este adevărată, falsitatea lui 3 este provocată de falsitatea lui –p. Prin urmare, p este adevărată. Cum clauza 1 conţine pe p şi q care sunt adevărate şi clauza 1 este falsă, falsitatea lui 1 este provocată de falsitatea lui –r. Dacă –r este fals, atunci, datorită principiului bivalenţei, r este adevărat. La acelaşi rezultat vom ajunge prin rezoluţia duală din S1* şi prin aplicarea regulilor dualităţii (vezi schemele de inferenţă R7, R1 şi R4).

$$\text{R7}\quad \frac{\models S1 = 1}{\models -(S1^*) = 1}$$

Derivăm rezolutiv dual din $S1^* = 0 \models_M -r = 0$:



Demn de reţinut este faptul că, în rezoluţia duală, conceptul central este cel de disjuncţie falsă. Dacă disjuncţia dintre clauzele 3 şi 18 este falsă, atunci sunt falşi ambii ei termeni. Prin urmare, -s este falsă şi, deci, potrivit legii negaţiei s este adevărată. Atunci falsitatea conjuncţiei 3 se datorează falsităţii lui –p. Scriem, deci, 21, după care –p este fals. Căutăm o conjuncţie elementară care să conţină pe opusul lui –p. Găsim

pel. Şi raţionamentul se repetă. Disjuncţia dintre 21 şi 1 trebuie să fie falsă. Dacă-i fals –p, atunci este adevărat p. Deci falsitatea lui 1 se va întemeia pe falsitatea lui q-r. Scriem 22.q-r. În final ajungem la 25, după care este fals –r. Deci r este adevărat.

Ne întoarcem la demonstrarea lui R10.

R10 afirmă că dacă din S derivă rezolutiv Robinson D, atunci din negaţia dualei lui S derivă resolutiv Maslov negaţia dualei lui D. Schematic:

R12. Dacă $S \models_R D$, atunci $-S^* \models_M -D^*$

Demonstrăm din ipoteză consecventul tezei de demonstrat, folosind, desigur, alte legi şi teoreme logice .

1. $S \models_R D$ ip

2. $S \models D$

Formula 2 se obţine din 1. După teorema derivării rezolutive, conform căreia orice derivare rezolutivă este o consecinţă logică semantică. Pe de altă parte, aplicând teorema deducţiei la 2, obţinem:

3. $S \supset D = T$

după care $S \supset D$ este o tautologie.

Aplicând la 3 regula de dualitate R6 obţinem:

4. $D^* \supset S^* = T$

Aplicând la primul termen al aseţiunii 4 legea contrapoziţiei, putem scrie:

5. $- S^* \supset - D^* = T$

Formula 5 poate fi rescrisă ca:

6. $[(S^* = 0) \supset (D^* = 0)] = T$

La rândul ei, 6 poate fi rescrisă ca:

7. $(S^* = 0) \models D^* = 0$

Pe baza legii negaţiei, 7 poate fi rescrisă ca:

8. $- S^* \models - D^*$

În sfârşit, dacă asupra setului iniţial de clauze S s-a aplicat o derivare rezolutivă Robinson, atunci S a fost o formă normală conjunctivă şi perechile de clauze au fost clauze disjunctive. În mod necesar, duala lui S , S*, va fi o formă normală disjunctivă şi asupra ei nu se poate aplica decât o derivare rezolutivă Maslov sau o rezoluţie duală. În consecinţă, vom scrie la indicele inferior al lui $\models$ pe M, $\models_M$.

Obţinem astfel:

9. $- S^* \models_M - D^*$ Q. E. D.

Concluzii

Spre deosebire de rezoluția clasică creată de Robinson, rezoluția duală introdusă de Maslov nu este un instrument de extorcare a consecințelor logice valide dintr-un set de premise sau dintr-o bază de cunoștințe. Ea descrie o cale de înaintare de la o concluzie dată la posibilele ei temeiuri sau suporturi. Primul gând care ne trece prin minte este întrebarea dacă tehnicile ei nu pot fi cumva utilizate în teoria argumentării sau în căutarea textelor demonstrative.

Rezoluția duală permite trecerea de la falsitatea clauzelor-părinți la falsitatea rezolventului sau de la falsitatea setului inițial de clauze disjunctive, la falsitatea derivărilor rezolutive duale.

Am văzut mai sus relevanța teoriei rezoluției duale pentru problema minimizării funcțiilor logice, ca și pentru studiul diviziunilor și clasificărilor. Nutrim convingerea că rezoluția duală este importantă pentru relansarea teoriei conceptelor, a definițiilor și clasificărilor. Asupra unora dintre aceste probleme vom reveni într-un capitol următor din volumul II, consacrat teoriei clasificărilor.

CAP. 7. METODA EXTINDERII CONSERVATIVE A INTERPRETĂRILOR PARŢIALE

Fie un set de clauze disjunctive legate prin conjuncţie sau o formă normală conjunctivă (FNC) a unei formule α. O formulă normală conjunctivă are structura:

$\alpha = \wedge ki, i \geq 0$

$ki = \vee lj, j \geq 0$

unde ki sunt clauze disjunctive iar lj sunt literali.

Exemplu 1

$$\alpha_1 = (p \vee -q) \wedge (p \vee r) \wedge (-p \vee s) \wedge (-p \vee -q \vee -r) \wedge (q \vee -s) \wedge p$$
$$\quad 1 \qquad\quad 2 \qquad\quad 3 \qquad\quad 4 \qquad\qquad 5 \quad 6$$

Când este realizabilă o formulă având structura α1?. Evident, când sunt adevăraţi toţi termenii conjuncţiei, i.e. toate clauzele disjunctive, iar o clauză disjunctivă este adevărată, dacă este adevărat cel puţin un literal din ea.

Un set de clauze ce conţine o clauză unitară (o clauză alcătuită dintr-un singur literal) este realizabil dacă literalul respectiv ia valoarea 1. În exemplul de mai sus, α_1 este realizabil, dacă şi numai dacă k6 = p = 1, unde k6 este o clauză unitară. Dacă p = 1, atunci k6 = k1 = k2 = 1. Pe această cale putem trage concluzia că problema realizabilităţii formulei α_1 se reduce la problema realizabilităţii formulei α_2:

$$\alpha_2 = (-p \vee s) \wedge (-p \vee -q \vee -r) \wedge (q \vee -s)$$

Trecerea de la α_1 la α_2 s-a făcut pe baza regulilor deductive: Dacă $A \wedge 1$, atunci A sau altfel scrisă:

$$\frac{A \wedge 1}{A}$$

Deci, un model sau o interpretare care satisface setul de clauze α1 va trebui să conţină, în mod necesar, p = 1. Deci, până acum avem: $\models \{p,...\}$

Mai departe, potrivit principiului bivalenţei şi definiţiei negaţiei, dacă p = 1, atunci -p = 0. Deci, α_2 poate fi rescrisă ca:

$$\alpha_3 = (0 \vee s) \wedge (0 \vee -q \vee -r) \wedge (q \vee -s)$$

Deoarece $0 \vee A = A$, putem scrie pe α_3 ca:

$$\alpha_4 = s \wedge (-q \vee -r) \wedge (q \vee -s)$$

Dar α_4 conţine clauza unitară s. În consecinţă, pe acelaşi temei, putem determina încă o condiţie necesară a veridicităţii setului de clauze α. Pentru ca setul de clauze α să fie adevărat într-o interpretare I, aceasta trebuie să satisfacă s =1. Adăugăm, deci I = {p, s,...}. Substituind s cu 1 în α_4, obţinem:

$$\alpha_5 = (-q \vee -r) \wedge q$$

Mai departe, pentru ca α_5 să fie adevărată, trebuie ca q să fie adevărată, i.e. q = 1. Am mai făcut încă un pas în găsirea unui model pentru setul iniţial de clauze, α. Deci, I = {p, s, q,...}. Înlocuind pe q prin valoarea sa, i.e. q = 1, din α_5 obţinem:

$$\alpha_6 = (0 \vee -r)$$

de unde rezultă:

$$\alpha_7 = -r,$$

respectiv, r = 0.

Punând cap la cap toate condiţiile necesare pentru veridicitatea lui α_1 identificate mai sus, putem conchide că o interpretare ce satisface pe α_1 este I = {p, s, q, -r } sau p = q = s = 1 şi r = 0.

Exemplul analizat mai sus ne sugerează faptul că se pot formula reguli de reducere succesivă a realizabilităţii unei formule mai complexe la realizabilitatea unor formule mai simple, până când, în cele din urmă, găsim o atribuire de valori care satisface formula iniţială. Metoda Davis-Putnam este o colecţie de astfel de reguli, care se sprijină, desigur, pe anumite legi logice sau teoreme.

Să mai cercetăm încă un exemplu.

Exemplul 2

$$\beta = (p \vee -q \vee r) \wedge (-p \vee s) \wedge (-s \vee t) \wedge (q \vee u) \wedge (-u \vee t) \wedge (-r \vee v) \wedge (-v \vee t) \wedge -t$$
$$\quad 1 \qquad\quad 2 \qquad\quad 3 \qquad\quad 4 \qquad\quad 5 \qquad\quad 6 \qquad\quad 7 \qquad\quad 8$$

Convenim să transcriem fiecare clauză ca o mulţime de literali şi să le ordonăm alfabetic pe coloane.

	p	q	r	s	t	u	v
k1	p	-q	r				
k2	-p			s			
k3				-s	t		
k4		q				u	
k5					t	-u	
k6			-r				v
k7					t		-v
k8					-t		

Tabloul nr. 1

Clauza k_8 = -t este o clauză unitară din β. Pentru ca β să fie realizabil trebuie să existe o interpretare I în care t = 0, căci numai astfel k_8 = 1, i.e. I = {-t,...}.Dacă t = 0, atunci adevărul clauzelor k_7, k_5, k_3, trebuie să se întemeieze pe alţi literali decât t, căci t = 0. În consecinţă, pentru k_7 I trebuie să conţină v = 0, ca -v să fie adevărată, căci numai astfel k_7 este adevărată şi fără adevărul lui k_7 nu poate fi adevărată nici β. Din aproape în aproape, pe această cale, obţinem noi determinări sau condiţii necesare pentru interpretarea I a lui β, I = {-t, -v, -u, -s,...}. Dacă v = u = s = 0 , atunci, pe temeiuri similare cu cele dintr-un raţionament anterior, pentru adevărul lui k_6, k_5, şi k_3, trebuie să fie adevărate -r, q, şi -p, respectiv să determinăm pentru I, r = 0, q = 1 şi p = 0. Astfel obţinem o interpretare I = {-t, -v, -u, -r, q, -p} care satisface, în ordine, pe k_8, k_7, k_5, şi k_3 şi în sfârşit, pe k_6, k_4, k_2, deci clauzele k_2-k_8, dar care nu satisface k_1. Într-adevăr, pentru p = r = 0 şi q = 1, k_1 = 0. Dacă, dimpotrivă, dorim să facem pe k1 adevărată, atunci vom obţine valorizări şi modele care nu vor satisface clauzele k_2-k_8. Setul de clauze β este irealizabil.

Şi în tentativa de mai sus de a găsi un model sau o interpretare am pornit tot de la clauza unitară, care impunea anumite constrângeri în interpretarea lui β; prima condiţie a generat alte constrângeri care s-au propagat mai departe. La capătul drumului se impunea, pentru ultima clauză luată în considerare, un set de valori care o făceau în mod necesar falsă: $k_2 \wedge k_3 \wedge ... \wedge k_8 \supset -k_1$. Dar aceasta ilustrează însăşi ideea de incompatibilitate, $-k_2 \vee -k_3 \vee ... \vee -k_8 \vee$ -k1.

Exemplele de mai sus le putem generaliza într-o metodă de decizie aplicabilă la seturi de disjuncţii elementare sau la forme normale conjunctive. Fie α o astfel de formulă.

Metoda poate fi descrisă, într-o primă aproximaţie, prin următoarele reguli sau instrucţiuni:

I1.Scrie pe prima linie, în ordine alfabetică, variabilele care apar în formula α;

I2. Scrie literalii din fiecare clauză, alfabetic, sub literele corespunzătoare din alfabetul formulei α;

I3.Identifică o clauză unitară (pozitivă sau negativă) din α şi valorizarea care o face adevărată (dacă există) şi trece literalul clauză unitară în interpretarea I;

I4. Taie, în tabel, literalii opuşi acesteia care apar în celelalte clauze, literali aflaţi pe aceiaşi coloană cu clauza unitară;

I5.Identifică între clauzele modificate la punctul anterior clauzele unitare nou apărute şi aplică-le şi lor instrucţiunea 3.

I6. Dacă s-a a ajuns ca interpretarea I să facă adevărate toate cele n clauze, atunci I este un model al lui α şi α este realizabilă. Dimpotrivă, dacă interpretarea I nu poate fi extinsă astfel încât să cuprindă toate cele n clauze, atunci α este irealizabilă.

Metoda aplicată mai sus utilizează, în esenţă, definiţiile conjuncţiei şi disjuncţiei şi unele legi logice de forma: $A \wedge 1 = A$, $0 \vee A = A$, $((A \vee B) \wedge \neg A) \supset B$, legi admise curent în logica clasică si utilizate şi în metoda rezoluţiei sau în metoda Davis-Putnam.

Procedeul descris mai sus este adecvat pentru cazurile când în α (sau în setul de clauze S) apar clauze unitare, iar în alte clauze, neunitare, apar opuşii acestora. El conduce la o simplificare importantă a setului iniţial de clauze. Ce se întâmplă, însă, dacă în setul iniţial S nu apare nici o clauză unitară ? Atunci înstrucţiunile de mai sus nu sunt suficiente şi trebuie completate cu instrucţiunile:

I7. Alege un literal *lit* (eventual cu frecvenţa cea mai mare) din setul iniţial S, ce nu conţine clauza unitară sau din setul derivat S', obţinut din aplicarea instrucţiunilor 1-6 şi trece-l în interpretarea I;

I8. Elimină din setul S (curent) clauzele ce conţin literalul *lit* şi redu, astfel pe S la un subset al acestuia, S' (această reducere poate fi redată grafic printr-o bară verticală îngroşată, trasă după apariţia literalului *lit* în clauzele ce-l conţin);

I9. Elimină din clauzele lui S' literalii opuşi lui *lit,* respectiv literalii *-lit* ;

I10. Aplică la S' instrucţiunea I7 până când S' se epuizează, i.e. S' = Φ. Dacă S' = Φ, atunci S este realizabil şi interpretarea I astfel obţinută este un model al setului iniţial de clauze. Dacă, dimpotrivă, în

analiza lui S, *ajungem*, aplicând regulile I1-I9, *să scriem pe o coloană o pereche opusă de literali, (lit, -lit), atunci setul iniţial S este inconsistent*, deoarece conduce la atribuirea unuia şi aceluiaşi literal şi a valorii de adevăr 1 şi a valorii de adevăr 0. În acest caz, formula iniţială S este irealizabilă.

Convenim să numim metoda prezentată mai sus drept metoda extinderii conservative a interpretărilor semantice parţiale. Am numit-o anterior metoda determinărilor semantice succesive.

Metoda prezentată mai sus este compatibila cu o tratare computerizată a problemelor şi poate fi făcută relevantă pentru analiza şi întemeierea logică a unor decizii economice sau tehnice.

Exemplul 3

Se dă situaţia acţională descrisă de următoarele 8 propoziţii din limba naturală. Să se identifice caracteristicile deciziei sau soluţiei problemei puse de această situaţie acţională.

1. *Dacă afacerile îmi merg bine, îmi construiesc o vilă.*
2. *Dacă îmi construiesc o vilă o voi ridica la Costineşti, la Predeal sau la Snagov.*
3. *Dacă îmi construiesc o vilă o voi face din cărămidă sau din prefabricate.*
4. *Dacă vila e construită din prefabricate vor emana în interior substanţe toxice.*
5. *Dar vila mea nu va emana în interior substanţe toxice.*
6. *Afacerile îmi merg bine.*
7. *Dacă îmi ridic vila la Costinesti sau la Predeal va fi departe de casa mea din Bucureşti.*
8. *Dar vila mea nu va fi departe de casa mea din Bucureşti.*

Întrebări: *a. Unde îmi construiesc eu vila?; b. Din ce materiale o construiesc?*

Formalizăm în limbajul logicii propoziţiilor situaţia actională descrisă în propoziţiile de mai sus:

1' $p \supset q$

2'. $q \supset (r \vee s \vee t)$

3'. $q \supset (r1 \vee s1)$

4' $s1 \supset v1$

5' $-v1$

6. p

7. $(s \vee t) \supset d$
8. -d

Formalizăm întrebările:

a. ?- $(r \vee s \vee t)$; b. ?- $(r1 \vee s1)$

Aducem formulele de mai sus la formă clauzală:

1" $-p \vee q$
2". $-q \vee r \vee s \vee t$
3". $-q \vee r1 \vee s1$
4". $-s1 \vee v1$
5" $-v1$
6". p
7.1". $-r \vee d$
7.2". $-s \vee d$
8.". -d

Constituim setul de clauze iniţiale ce descrie situaţia acţională sau datele problemei logice.

$S3 = (-p \vee q) \wedge (-q \vee r \vee s \vee t) \wedge (-q \vee r1 \vee s1) \wedge (-s1 \vee v1) \wedge -v1 \wedge p \wedge$
$\wedge (-r \vee d) \wedge (-s \vee d) \wedge -d$

Am obţinut astfel setul iniţial de clauze sau forma normală conjunctivă care descrie datele problemei logice al cărei model sau soluţie trebuie să o descoperim prin utilizarea metodei interpretărilor semantice succesive prezentată mai sus.

Construim acum, în conformitate cu instrucţiunile 1 şi 2, tabloul asociat setului S3.

	p	q	r	s	t	r1	s1	v1	d
k1	-p	q							
k2		-q	r	s	t				
k3		-q				r1	s1		
k4							-s1	v1	
k5								-v1	
k6	p								
k7.1			-r						d
k7 2				-s					d
k8									-d

Tabloul T3

Aplicând la tabloul T3 instrucţiunea I3, alegem din T3 clauza unitară 6 şi construim interpretarea parţială Intl = {p, ...}, i.e. atribuim lui p valoarea 1 în interpretarea I, adevărul propoziţiei p fiind o condiţie necesară pentru adevărul lui S3. Mai departe , pe baza lui $A \wedge 1 = 1$, reducem setul de clauze S3 la S' = S3 \ k6, unde k6 = p. În consecinţă, vom rescrie tabloul 3 fără linia 6 Totodată, în conformitate cu instrucţiunea I4, vom tăia din kl pe -p, căci -p = 0, dacă p = 1 şi ,într-o disjuncţie $A \vee 0 = A$. După tăierea liniei kl şi a coloanei 2 rescriem, mai jos, T3 ca T4:

	q	r	s	t	rl	sl	vl	d
kl	q							
k2	-q	r	s	t				
k3	-q				rl	sl		
k4						-sl	vl	
k5							-vl	
k7.1		-r						d
k7.2			-s					d
k8								-d

Tabloul T4

Mai departe, kl din T4 este o clauză unitară. Aplicăm la T4 instrucţiunile I3 şi I4 completăm interpretarea parţială I1 cu literalul q şi obţinem Int 2 = {p, q,...} şi apoi tăiem linia kl şi coloana 2 din T4 şi obţinem tabloul T5:

	r	s	t	rl	sl	vl	d
k2	r	s	t				
k3				rl	sl		
k4					-sl	vl	
k5						-vl	
k7.1	-r						d
k7.2		-s					d
k8							-d

Tabloul T5

Singura clauză unitară în T5 este k8. Din nou, pe baza instrucţiunilor I3 şi I4, obţinem Int 3 = {p, q, -d, }. Mai departe, tăind linia k8 şi coloana 8 obţinem tabloul T6:

	r	s	t	r1	s1	v1
k2	r	s	t			
k3				r1	s1	
k4					-s1	v1
k5						-v1
k7.1	-r					
k7.2		-s				

Tabloul T6

Clauze unitare în T6 sunt k7.1, k7.2 şi k5. Selectăm clauza k7.2 şi aplicăm instrucţiunile I3 şi I4. Pe baza instrucţiunilor I3 şi I4, obţinem:

Int4 = {p, q, -d, -s . . . } şi tabloul T7:

	r	t	r1	s1	v1
k2	r	t			
k3			r1	s1	
k4				-s1	v1
k5					-v1
k7.1	-r				

Tabloul T7

Clauze unitare în T7 sunt k7.1 şi k5. Selectăm pe k7.1. Scriem Int 5 = {p, q, -d, -s, -r }, ştergem linia k7.1 şi coloana 2 şi obţinem tabloul T8:

	t	r1	s1	v1
k2	t			
k3		r1	s1	
k4			-s1	v1
k5				-v1

Tabloul T8

În tabelul T8 sunt două clauze unitare: k2 şi k5. Aplicăm instrucţiunile I3 şi I4 la clauza k5. Obţinem interpretarea In 6 = {p, q, -d, -s, -r, -v1 } şi tabloul T9:

	t	r1	s1
k2	t		
k3		r1	s1
k4			-s1

Tabloul T9

Tabelul T9 conţine clauzele unitare k2 şi k4. Aplicăm instrucţiunile I3 şi I4 la k4. Obţinem interpretarea
Int7 = {p, q, -d, -s, -r, -v, -s1 } şi tabloul T10:

	t	r1
k2	t	
k3		r1

Tabloul T10

Pe baza aceloraşi instrucţiuni scriem interpretarea
Int8 = {p, q, -d, -s, -r, -v, -s1, r1 . . } şi eliminăm din tabloul T10 linia k3 şi coloana 3 şi obţinem tabloul T11:

	t
k2	t

Tabloul T11

Identificăm pentru T11 interpretarea Int 9, respectiv interpretarea finală sau soluţia problemei, care este
Int 9 = {p, q, -d, -s, -r, -v1, -s1, r1, t}şi tăiem linia k2 şi coloana 2 obţinând:

□

T12 = Φ

Soluţia problemei este p∧q∧-d∧-s∧-r∧-v1∧-s1∧r1∧t, care redată în limba română, conform convenţiilor iniţiale de codificare (vezi începutul exemplului 3) ne informează că: afacerile îmi merg bine, îmi construiesc o vilă, nu departe de casa din Bucureşti, nu la Predeal şi nu la Costineşti, în interiorul căreia nu vor emana substanţe toxice, nu din prefabricate, din cărămidă, la Snagov.

Propunem cititorului dornic de exercițiu să verifice soluția probemei date prin: metoda schemelor de inferență sau prin calcul natural, prin aplicarea metodei rezoluției la setul de clauze S3 sau prin metoda Davis-Putnam. Îi sugerăm, de asemenea, construirea unei probleme similare cu date din viața economică sau din activitățile social-politice.

Observații

1. A afla soluția unei probleme redată în limbajul logicii propozițiilor, în particular sub formă de set de clauze, înseamnă a găsi o interpretare sau o atribuire de valori dată variabilelor în care toate datele, clauzele sau restricțiile problemei să rămână adevărate;

2. Fiecare clauză sau condiție descrie un spațiu de stări posibile sau alternative în care se caută valoarea acordată variabilelor problemei. Rezolvarea pas cu pas a problemei înseamnă a elimina unele alternative, a restrânge spațiul de căutare, a determina, treptat, valorile variabilelor;

3. Clauzele unitare nu au alternative. În cazul lor, valoarea variabilei este determinată univoc; ele descriu restricții strict determinate;

4. Spre deosebire de ecuațiile din algebră, care iau valori din domeniile numerelor, în problemele de logică formulele iau valori în domeniul valorilor de adevăr admise într-o teorie logică sau alta. În logica clasică, acestea sunt 1 şi 0. Dar aici se face adesea convenția după care acceptarea ca adevărată a unei formule înseamnă pur şi simplu a o scrie, iar respingerea ei înseamnă a o scrie precedată de semnul negației;

5. Apropierea de soluția finală s-a făcut prin determinări semantice succesive, respectiv prin descoperirea prin regulile de inferență a asignărilor de valori impuse de datele inițiale;

6. Valorizarea unui literal într-o clauză a atras de fiecare dată consecințe imediate în privința valorizării altor literali din celelalte clauze, în deosebi a opuşilor lor (prin proprietățile operatorului negației) sau a clauzelor în care apar aceştia. La rândul ei, valorizarea unei clauze are consecințe asupra condițiilor de adevăr ale întregului set de clauze;

7. Metoda utilizată nu identifică *toate* soluțiile problemei logice, ci o singură soluție, un singur model sau interpretare ce satisface setul de clauze, respectiv datele au restricțiile problemei. În scrierea adoptată caracterul de ecuație logică al setului de clauze S

este mascat. Noi scriem, pur și simplu, S sau forma normală conjunctivă a enunțului problemei, în loc să scriem:

$$v(S) = 1 \qquad (a)$$

și apoi să explicităm aceasta scriind:

$$v(S) = 1 \supset \forall k\,(k \in S \supset v(k) = 1) \qquad (b)$$

$$v(k) = 1 \supset \exists \ell\,(\ell \in k \wedge v(\ell) = 1) \qquad (c)$$

respectiv:

Dacă un set de clauze S este adevărat, atunci toate clauzele sale $k \in S$ sunt adevărate (b).

Dacă o clauză este adevărată, atunci cel puțin un literal ℓ din k este adevărat (c).

În mod analog, ar trebui să scriem explicit condițiile de fals ale setului S:

$$v(S) = 0 \supset \exists k\,(k \in S \wedge v(k) = 0) \qquad (d)$$

$$v(k) = 0 \supset \forall \ell\,(\ell \in k \supset v(\ell) = 0) \qquad (e)$$

respectiv:

O problemă nu are soluție, dacă cel puțin o restricție a sa nu este satisfăcută;

O restricție sau o clauză a unei probleme nu este satisfăcută, dacă toți literalii ei sunt falși.

8. Rezolvarea problemei a constat într-o serie de reduceri succesive a spațiului de căutare, într-o serie de reformulări a restricțiilor și în *determinarea*, pas cu pas, a valorii de adevăr a *tuturor* variabilelor. Or, potrivit principiului verifuncționalității, a deține o atribuire de valori pentru *toate* variabilele (respectiv *toți atomii*), înseamnă a putea calcula valorile de adevăr pentru toate formulele compuse derivabile.

Reducerile au constat în eliminarea clauzelor pentru care s-au găsit evaluări concordante cu asumările anterioare și care le fac adevărate. Grafic, eliminarea unei clauze s-a exprimat prin tăierea liniei din tabel pe care a fost scrisă aceasta și prin tăierea coloanei în care a fost scrisă variabila sau literalul adăugat în interpretarea I. În acest caz, soluția

problemei depindea, în exclusivitate, de găsirea unor evaluări convenabile pentru *celelalte* variabile şi respectiv pentru celelalte clauze.

9. Problema din exemplul 3 a avut toţi paşii determinaţi sau coercitivi şi a fost rezolvată apelând, exclusiv, la primele patru instrucţiuni, I1-I4. Problemele care fac apel la instrucţiunile I5, I6, I7 au soluţii alternative şi angajează mai mult iniţiativa şi fantezia rezolvitorului;

10. Credem oportună descrierea demersurilor în rezolvarea unei probleme cu ajutorul unor arbori, în care putem marca paşi liniari sau constrângători şi punctele de bifurcare sau de împrăştiere, nodurile de „iniţiativă" ale rezolvitorului, unde acesta emite o ipoteză sau face o evaluare temporară etc;

11. Sub aspect grafic, pot fi găsit scrieri mai concise. Tăierea unei clauze poate fi marcată printr-un simbol special şi, la fel, tăierea unei coloane, astfel încât întreaga rezolvare să se facă pe un singur tabel. În fapt, esenţială este identificarea listei de literali ce descrie valorizarea variabilelor sau soluţia problemei. Important este să găsim valorizările impuse de restricţiile problemei, să satisfacem exigenţele fiecăreia dintre clauzele acesteia, să „acoperim" prin valorizări adecvate clauzele.

Cap. 8. METODA DAVIS-PUTNAM

Ca şi metoda rezoluției, şi metoda determinărilor semantice succesive, metoda Davis-Putnam se aplică la seturi de clauze disjunctive sau la forme normale conjunctive pentru a verifica realizabilitatea acestora. Metoda este cu prioritate o tehnică de simplificare şi de decizie şi numai indirect şi în secundar o metodă de deducție. Desigur, într-un anumit sens putem prezenta metoda Davis-Putnam ca o metodă deductiv-reductivă: prin ea reducem o problemă mai complexă la una sau două probleme mai simple. Mai exact, reducem problema realizabilității unui set de clauze mai complex la realizabilitatea unuia sau la realizabilitatea altor două seturi de clauze mai simple. Ea poate fi văzută şi ca o tehnică de căutare sistematică a unui model pentru un set de clauze.

1 Regulile metodei Davis-Putnam

Regula clauzei unitare (RCU)

Dacă într-un set de clauze apare o clauză unitară k, $k = \ell$, unde ℓ este un literal, construim, pornind de la S, un set S1 prin eliminarea din S a clauzei unitare şi a tuturor clauzelor din S ce conțin literalul clauză unitară. Dacă $S1 = \Phi$, atunci S este realizabil. Dacă $S \neq \Phi$, atunci construim, pornind de la S1, un nou set de clauze S2, prin eliminarea din S1 a tuturor literalilor identici cu negația literalului clauză unitară; setul inițial de clauze S este realizabil, dacă şi numai dacă este realizabil setul S2.

Regula literalului pur (RLP)

D1. Numim *literal pur* în setul S de clauze un literal a cărui negație nu apare în S.

Fie S un set de clauze ce conține un literal pur. Setul S este realizabil, dacă şi numai dacă este realizabil un set de clauze S1 obținut din S prin eliminarea tuturor clauzelor ce conțin literalul pur.

Regula clauzei tautologice(RCT)

Dacă un set de clauze S conţine clauze tautologice se construieşte un set S1, identic cu S, cu singura deosebire că din el s-au eliminat toate clauzele tautologice. Setul S este realizabil, dacă şi numai dacă setul S1 este realizabil.

Regula clauzei consecinţă(RCC)

Dacă un set de clauze S conţine două clauze, k1 şi k2, astfel că $k1 \supset k2$ se construieşte un set de clauze S1 identic cu S cu singura deosebire că din el s-a eliminat clauza consecinţă k2. S este realizabil, dacă şi numai dacă este realizabil S1.

Regula scindării(RS)

D2. Fie S un set de clauze şi ℓ un literal în S al cărui opus apare şi el în S. Partiţionăm atunci mulţimea de clauze S, după criteriul literalului ℓ în trei submulţimi:

D2.1. $S_\ell^+ = \{ k \in S: \ell \in k\}$

D2.2. $S_\ell = \{ k \in S: -\ell \in k\}$

D2.3. $S_\ell^0 = \{ k \in S: \ell \notin k \wedge -\ell \notin k\}$

Pe baza definiţiilor de mai sus introducem:

D2.4 $POS_\ell(S) = S_\ell^0 \cup \{k\setminus \{\ell\}: k \in S_\ell^+\}$

D2.5. $NEG_\ell(S) = S_\ell^0 \cup \{k\setminus \{-\ell\}: k \in S_\ell^-\}$

Formulăm acum regula scindarii.

Dacă S este un set de clauze în care apare literalul ℓ, atunci construieşte un set $S1 = POS_\ell(S)$ şi un set $S2 = NEG_\ell(S)$. Setul S este realizabil, dacă şi numai dacă este realizabil cel puţin unul dintre seturile S1 sau S2.

În metoda Davis-Putnam se recomandă aplicarea cu prioritate a regulii clauzei unitare şi literalului pur, a clauzei tautologice şi clauzei consecinţă şi numai în ultimă instanţă, a regulii scindării.

2. Teorema scindării (Splitting Rule)

Înainte de a ilustra aplicarea metodei Davis-Putnam şi de a demonstra teorema scindării este util să facem mai bine înţelese conceptele de $POS_\ell(S)$ şi $NEG_\ell(S)$. Fiind dat un set de clauze S. potrivit definiţiilor D2.1-D2.3, S_ℓ^+ desemnează submulţimea clauzelor din S în care apare literalul ℓ, S_ℓ denumeşte submulţimea clauzelor din S în care apare literalul $-\ell$, iar S_ℓ^0 se referă la submulţimea clauzelor dın S în care nu apare nici literalul ℓ, nici literalul $-\ell$.

Fie $S = (p \vee q \vee -r) \wedge (p \vee -s) \wedge (-p \vee t) \wedge (r \vee s) \wedge -t$.
Atunci:

$S_p^+ = \{p \vee q \vee -r,\ p \vee -s\}$ i.e. clauze care conţin literalul p

$S_p = \{-p \vee t\}$ i.e. clauze care conţin literalul -p

$S_p^0 = \{ r \vee s, -t\}$ i.e. clauze care nu conţin nici p, nici -p, „neutre"

$S1 = POS_p(S) = \{ r \vee s, -t, q \vee -r, -s\}$

$S2 = NEG_p(S) = \{ r \vee s, -t, t\}$

Mulţimea de clauze $POS_p(S)$ conţine submulţimea clauzelor „neutre" în raport cu literalul p, respectiv clauzele din S care nu conţin nici literalul p, nici literalul –p, la care s-au adăugat clauzele ce-l conţin pe p, după ce s-a omis din fiecare clauză literalul p.

Mulţimea de clauze $NEG_p(S)$ conţine submulţimea clauzelor „neutre" în raport cu literalul p, la care s-au adăugat clauzele ce-l conţin pe -p, după ce s-a eliminat din fiecare dintre ele literalul -p.

Potrivit regulii scindării, setul de clauze S este realizabil, dacă şi numai dacă cel puţin una dintre mulţimile de clauze $POS_p(S)$ sau $NEG_p(S)$ este realizabilă. Întrucât $NEG_p(S)$ nu este realizabilă, deoarece conţine o pereche opusă de clauze unitare, t şi -t, S va fi realizabil, dacă şi numai dacă $POS_p(S)$ este realizabil.

Testăm realizabilitatea setului $S1 = POS_p(S)$:

$S1 = \{ r \vee s, -t, q \vee -r, -s\}$

$S1a = \{r, -t, q \vee -r\}$ $(RCU_{-s}, S1)$

$S1b = \{-t, q\}$ $(RCU_r\ S1a)$

$S1c = \{q\}$ $(LP_{-t}, S1b)$

$S1d = \varnothing$ $(LP_q, S1c)$

Mulţimea S de clauze este realizabilă pentru interpretarea $I = \{-p, q, r, -s, -t\}$.

Geneza conceptelor $POS_I(S)$ şi $NEG_I(S)$ poate fi explicată cu ajutorul noţiunii de interpretare parţială.

Dăm pentru setul S, considerat mai sus, interpretarea parţială $I1 = \{p,\ \}$ sau $p = 1$. Atunci S devine:

1. $(1 \vee q \vee -r) \wedge (1 \vee -s) \wedge (-1 \vee t) \wedge (r \vee s) \wedge -t$.
2. $1 \wedge 1 \wedge (0 \vee t) \wedge (r \vee s) \wedge -t$ $(1, 1 \vee A = 1)$
3. $t \wedge (r \vee s) \wedge -t$ $(2,\ 0 \vee A = A)$
4. $\square$ $(3, A \wedge -A = 0, A = t)$

Interpretarea parţială I1 = {p, ...} a generat, aşadar, forma normală conjunctivă t ∧ (r∨ s) ∧ -t, care scrisă ca mulţime de clauze { t, r∨ s, -t } este tot una cu S2 = $NEG_p(S)$. În mod analog, $POS_p(S)$ va fi generat de interpretarea parţială I2 = {-p, . . . } sau p = 0:

1. (0 ∨ q∨ -r) ∧ (0 ∨ -s) ∧ (-0∨ t) ∧ (r∨ s) ∧ -t
2. (q∨ -r) ∧ -s ∧ (1∨ t) ∧ (r∨ s) ∧ -t (1, 0∨ A = A)
3. (q∨ -r) ∧ -s ∧ 1∧ (r∨ s) ∧ -t (2, 1 ∨ A = 1)
4. (q∨ -r) ∧ -s ∧ (r∨ s) ∧ -t (3, A∧1 = A)

Dacă aplicăm la 4 legea comutativităţii conjuncţiei şi transcriem forma normală conjunctivă ca o mulţime de clauze obţinem setul de clauze:

S1 = POS_pS) = { r∨ s, -t, q∨ -r, -s}

De aici vom conchide că regula scindării are la bază două interpretări parţiale opuse a unui set de clauze oarecare. Atribuirea în interpretarea I1 literalului p a valorii 1, i.e. p = 1 ne conduce la setul de clauze $NEG_p(S)$, iar atribuirea în interpretarea I2 literalului p a valorii 0, i.e. p = 0 ne conduce la setul de clauze POS_pS). Cum însă într-o logică bivalentă un literal p nu poate lua decât valorile 1 sau 0, rezultă că dacă setul de clauze S este adevărat, atunci, în mod necesar, va fi adevărată una dintre mulţimile de clauze $NEG_p(S)$ sau POS_pS).

Admiţând că ℓ este un literal dintr-un set de clauze disjunctive legate între ele prin conjuncţie (FNC), putem să exprimăm partiţia lui S după criteriul ℓ astfel:

D1 $S_i^+ = \ell \vee B_i$ $0 \le i \le m$

D2 $S_\ell = -\ell \vee C_j$ $0 \le j \le n$

D3 $S_\ell^0 = R$

unde B_i şi C_j sunt clauze disjunctive ce nu conţin literalii ℓ şi $-\ell$. În această scriere $POS_\ell(S)$ şi $NEG_\ell(S)$ vor fi definite ca:

D4. $POS_\ell(S) = B_i \wedge R$ $0 \le i \le m$

D5. $NEG_\ell(S) = C_j \wedge R$ $0 \le j \le n$

D6. $S = (\ell \vee B_i) \wedge (-\ell \vee C)_j \wedge R$ $\le i, j \le m,n$

Putem trece acum la demonstrarea teoremei „scindării" sau teoremei Davis-Putnam.

T1. Teorema scindării. Fie ℓ un literal dintr-un set de clauze S. Atunci S este realizabil, dacă şi numai dacă, disjuncția dintre $POS_\ell(S)$ şi $NEG_\ell(S)$ este realizabilă.

Demonstrarea teoremei presupune:

a) Dacă S este realizabil, atunci disjuncția dintre $POS_\ell(S)$ şi $NEG_\ell(S)$ este realizabilă;

b) Dacă disjuncția dintre $POS_\ell(S)$ şi $NEG_\ell(S)$ este realizabilă, atunci S este realizabil.

Demonstrăm, mai întâi, implicația de la stânga la dreapta.

1) S	ip.
2) $(\ell \vee B_1) \wedge (-\ell \vee C_1) \wedge R$	RE, 1), D6
3) $\ell \vee B_1$	E$\wedge$, 2)
4) $-\ell \vee C_1$	E$\wedge$, 2)
5) R	E$\wedge$, 2)
6) $\ell = 0$	ip. supl.
7) B_1	$0 \vee A = A$, 6), 3)
8) $B_1 \wedge R$	I$\wedge$, 7), 5)
9) $POS_\ell(S)$	RE, 8), D4
10) $\ell = 1$	ip. supl.
11) $-\ell = 0$	-, 10) i. e. neg
12) C_1	$0 \vee A = A$, 11), 4)
13) $C_1 \wedge R$	I$\wedge$, 12), 5)
14) $NEG_\ell(S)$	RE,13), D5

Prin aceasta prima parte a demonstrației, a) se încheie. Dacă setul S este realizabil, atunci disjuncția dintre $POS_\ell(S)$ şi $NEG_\ell(S)$ este realizabilă.

Să demonstrăm acum implicația de la dreapta la stânga, respectiv b).

Emitem ipoteza că:

15) $POS_\ell(S) \vee NEG_\ell(S)$	ip.
15b1 $POS_\ell(S)$	ip1
16) $B_1 \wedge R$	RE, 15b, D4
17) B_1	E$\wedge$, 16)
18) R	E$\wedge$, 16)
19) $\ell \vee B_1$	$q \supset p \vee q$, 17)

Considerăm acum o interpretare I care satisface 15b1, i.e. $POS_1(S)$pe care o extindem la o interpretare I' în care variabilei ℓ i se atribuie valoarea 0:

20) $\ell = 0$	ip. supl.
21) $-\ell = 1$, i.e. , $-\ell$	neg, -, 20)
22) $-\ell \vee C_j$	I $\vee$, 21)
23) $(\ell \vee B_i) \wedge(-\ell \vee C_j) \wedge R$	I$\wedge$, 19, 22), 18)
24) S	RE, 23, D6

Admitem acum că adevărul lui 15) se datorează adevărului lui $NEG_I(S)$:

25) $NEG_I(S)$	ip.
26) $C_j \wedge R$	RE, 25), D5
27) C_j	E$\wedge$, 26)
28) R	E$\wedge$, 26)
29) $-\ell \vee C_j$	I $\vee$, 27)

Considerăm acum o interpretare I care satisface 22)$NEG_I(S)$pe care o extindem la o interpretare I' în care variabilei ℓ i se atribuie valoarea 1:

30) $\ell = 1$, i.e. ℓ	ip. supl.
31) $\ell \vee B_i$	I$\vee$, 30)
32) $(\ell \vee B_i) \wedge(-\ell \vee C_j) \wedge R$	I$\wedge$, 31), 29), 28)

Cu aceasta demonstrația teoremei scindării este încheiată. Teorema scindării cuprinde, ca un caz particular, teorema clauzei unitare.

T2. Teorema clauzei unitare. Dacă într-un set de clauze S apare o clauză unitară, i.e. $k = \ell$, şi ℓ apare în alte clauze, atunci din adevărul setului de clauze S rezultă adevărul clauzelor neutre, S_ℓ^0, şi al clauzelor obținute din mulțimea clauzelor ce-l conțin pe ℓ, S_ℓ^+, prin eliminarea din fiecare clauză a literalului l.

Ipoteza suplimentară pe care o adăugăm pentru a trece de la teorema scindării la cea a clauzei unitare este:

$\exists i((\ell \vee B_i) \wedge(-\ell \vee C_j) \wedge R \wedge 0 \leq i \leq m \wedge B_i = 0) \vee$

$\exists j((\ell \vee B_i) \wedge(-\ell \vee C_j) \wedge R \wedge 0 \leq j \leq n \wedge = C_j = 0)$

Cu alte cuvinte, admitem că există în S cel puțin o clauză k_i în care B_i este fals sau o clauză k_j în care C_j este fals și, deci, $k_i = \ell$ sau $k_j = -\ell$. În consecință, k_i este o clauză cu un singur literal, iar adevărul ei trebuie să se sprijine pe ℓ.

Fie S un set de clauze ce conține o clauză unitară ℓ. Atunci, în conformitate cu definițiile D1-D6, S va avea forma:

1) $\ell \wedge (\ell \vee B_i) \wedge (-\ell \vee C_j) \wedge R$

Realizarea lui 1) presupune, în mod necesar, $\ell = 1$. Dacă $\ell = 1$, atunci 1) devine:

2) $1 \wedge (1 \vee B_i) \wedge (-1 \vee C_j) \wedge R$

3) $1 \wedge 1 \wedge (-1 \vee C_j) \wedge R$ $\quad A \wedge 1 = A,\ 1 \vee B = 1$

4) $(0 \vee C_j) \wedge R$ $\quad -1 = 0$

5) $C_j \wedge R$ $\quad 0 \vee C = C$

6) $NEG_i(S)$ $\quad$ RE, 5), D5

În mod similar, dacă clauza unitară este $-\ell$ i.e.

1) $S = -\ell \wedge (\ell \vee B_i) \wedge (-\ell \vee C_j) \wedge R$,

atunci S se reduce la :

7) $B_i \wedge R$

de unde, prin regula substituirii echivalențelor, RE, și definiția D4 de mai sus, avem:

8) $POS_i(S)$

Or, etapele parcurse mai sus reproduc instrucțiunile clauzei unitare.

Teorema scindării conține ca un caz particular și teorema asociată regulii literalului pur.

T3. Teorema literalului pur. Dacă într-un set de clauze S apare literalul ℓ, dar nu apare opusul acestuia, $-\ell$ sau invers, atunci S este realizabil, dacă și numai dacă, este realizabil setul clauzelor neutre(notat mai sus prin S_ℓ^0 sau prin R).

Dacă ℓ este literalul pur, atunci din S lipsesc clauzele de forma $-\ell \vee C_j$, i.e. $S_j^- = \varnothing$:

9) $S = (\ell \vee B_i) \wedge R$

Dacă 1 = 1, atunci S este realizabil, dacă și numai dacă, R este realizabil, respectiv, dacă clauzele neutre sunt realizabile. În mod analog se demonstrează cazul literalului pur negat, situația când într-un set de clauze apare un literal negat, dar acesta nu apare în formă pozitivă. În acest caz, din S lipsesc clauzele de forma : $\ell \vee B_i$ iar setul va avea forma:

10) $S = (-\ell \vee C_j) \wedge R$

Regulile metodei Davis -Putnam pot fi transpuse ca reguli de inferenţă astfel:

$$\frac{\ell \wedge (\ell \vee B_i) \wedge (-\ell \vee C_j) \wedge R}{C_j \wedge R} \quad \text{RCU1}$$

$$\frac{-\ell \wedge (\ell \vee B_i) \wedge (-\ell \vee C_j) \wedge R}{B_i \wedge R} \quad \text{RCU2}$$

$$\frac{(\ell \vee B_i) \wedge R}{R} \quad \text{RLP1}$$

$$\frac{(-\ell \vee C_j) \wedge R}{R} \quad \text{RLP2}$$

$$\frac{S1 \wedge \blacksquare}{S1} \quad \text{RCT}$$

$$\frac{S1 \wedge B_i \wedge (B_i \supset C_j)}{S1 \wedge B_i} \quad \text{RCC}$$

$$\frac{(\ell \vee B_i) \wedge (-\ell \vee C_j) \wedge R}{(B_i \vee C_j) \wedge R} \quad \text{RS}$$

Exemplul 1

$S = (p \vee -q \vee r) \wedge (-p \vee s) \wedge (-s \vee t) \wedge (q \vee u) \wedge (-u \vee t) \wedge$
$\wedge (-r \vee v) \wedge (-v \vee t) \wedge -t$

Putem aplica regula clauzei unitare în literalul -t (vezi RCU2) obţinând:

$S1 = (p \vee -q \vee r) \wedge (-p \vee s) \wedge -s \wedge (q \vee u) \wedge -u \wedge (-r \vee v) \wedge -v$

I= {-t,. }

Aplicăm aceeaşi regulă la S1 în literalul -v şi obţinem:

$S2 = (p \vee -q \vee r) \wedge (-p \vee s) \wedge -s \wedge (q \vee u) \wedge -u \wedge -r$

I= {-t, -v,. . }

Aplicăm, în continuare, aceeaşi regulă la S2 în literalul -r. Rezultă:

$S3 = (p \vee -q) \wedge (-p \vee s) \wedge -s \wedge (q \vee u) \wedge -u$

I= {-t, -v, -r. . . }

Aplicând, mai departe, RCU2 la S3, în literalul -u ajungem la:

$S4 = (p \vee -q) \wedge (-p \vee s) \wedge -s \wedge q$

I= {-t, -v, -r, -u. . . }

La S4 putem aplica RCU1 în literalul pozitiv q. Obţinem:

$S5 = (p \wedge (-p \vee s) \wedge -s$

I= {-t, -v, -r, -u, q. . . }

La S5 se poate aplica RCU2 în literalul -s. De unde rezultă:

$S6 = p \wedge -p$ i.e.

Rezultatul concordă cu concluzia la care am ajuns anterior prin testarea aceluiaşi set de clauze prin metoda interpretărilor semantice succesive, setul S testat mai sus fiind identic cu setul β din paragraful anterior.

Exemplul 2

$$C = (p \vee q \vee s) \wedge (p \vee -q) \wedge (q \vee s) \wedge (-p \vee t) \wedge (-t \vee s) \wedge \\ \wedge (-s \vee v) \wedge (-v \vee u) \wedge (-s \vee -u)$$

Setul de clauze C nu conţine nici o clauză unitară. Prin urmare, la el nu se poate aplica regula (sau regulile) clauzei unitare RCU (vezi RCU1 şi RCU2). La setul C se poate aplica regula clauzei consecinţe. Clauza 3 implică clauza 1, i.e. clauza 1 este o consecinţă a clauzei 3, respectiv $(q \vee s) \supset (p \vee q \vee s)$. În consecinţă, putem tăia clauza 1:

$$C1 = (p \vee -q) \wedge (q \vee s) \wedge (-p \vee t) \wedge (-t \vee s) \wedge (-s \vee v) \wedge \\ \wedge (-v \vee u) \wedge (-s \vee -u) \qquad RCC(C)$$

La C1 nu putem aplica nici o altă regulă decât regula scindării, RS. Aplicăm regula RS în literalul p şi calculăm mai întâi pe $POS_p(C1)$:

$$C2 = -q \wedge (q \vee s) \wedge (-t \vee s) \wedge (-s \vee v) \wedge (-v \vee u) \wedge \\ \wedge (-s \vee -u) \qquad POS_p(C1)$$

La C2 se poate aplica regula clauzei unitare în literalul -q :

$$C3 = s \wedge (-t \vee s) \wedge (-s \vee v) \wedge (-v \vee u) \wedge (-s \vee -u) \quad RCU_{-q}(C2)$$

La C3 este aplicabilă, din nou, regula clauzei unitare. De data aceasta în literalul s:

$$C4 = v \wedge (-v \vee u) \wedge -u$$

A fost tăiată clauza unitară s şi clauza $-t \vee s$ pentru că ea conţine pe s şi devine adevărată când i se dă lui s valoarea 1 , i.e. când s = 1; de asemenea, a fost tăiat literalul opus clauzei unitare -s din clauzele ce-l conţin, pentru că acesta devine fals, iar falsul într-o clauză nu-i sporeşte şansele de realizare.

Aplicăm la C4 regula clauzei unitare în literalul -u şi obţinem:

C5 = v ∧ -v

o instanţiere a contradicţiei logice sau a idei de irealizabilitate, notată mai sus prin □

Întrucât $POS_p(C1)$ este irealizabil, pentru a determina realizabilitatea setului C, trebuie să testăm, în continuare, $NEG_p(C)$.

C6 = (q ∨ s) ∧ t ∧ (-t ∨ s) ∧ (-s ∨ v) ∧ (-v ∨ u) ∧ (-s ∨ -u)

Au fost reţinute, din setul de clauze C1, clauzele „neutre", în raport cu p şi clauzele ce-l conţin pe -p, dar din care s-a eliminat literalul -p.

La C6 se aplică regula clauzei unitare, RCU, în literalul t. Obţinem:

C7 = (q ∨ s) ∧ s ∧ (-s ∨ v) ∧ (-v ∨ u) ∧ (-s ∨ -u)

La C7 se aplică regula clauzei unitare în literalul s

C8 = v ∧ (-v ∨ u) ∧ -u

Au fost eliminate clauzele ce-l conţin pe s, iar din cele ce-l conţin pe -s a fost eliminat -s. La C8 aplicăm regula clauzei unitare în literalul -u şi obţinem:

C9. v ∧ -v

din nou o instanţiere a principiului contradicţiei. Conchidem, mai întâi, că $NEG_p(C)$ nu este realizabil. Cum însă, nici POS_p (C) nu este realizabil, tragem concluzia finală că setul C nu este realizabil.

Sugerăm cititorului să verifice realizabilitatea setului C prin metoda interpretărilor semantice succesive şi prin metoda respingerilor rezolutive. Cititorul poate găsi aceste exerciţii rezolvate în cartea noastră *Logica predicatelor*, Editura Hyperion, 1992

Metoda Davis-Putnam se întemeiază pe interpretări parţiale, pe asignări de valori acordate unor variabile propoziţionale sau unor atomi predicativi şi pe o partiţie în trei submulţimi de cazuri: 1. care satisfac literalul pozitiv; 2 care satisfac negaţia acestuia; 3 care nu satisfac nici literalul pozitiv şi nici negaţia acestuia. Metoda Davis-Putnam poate sta la baza unei teorii a clasificării obiectelor sau situaţiilor descrise în termenii atomilor predicativi care apar într-o bază de cunoştinţe.

Fiecare atom predicativ luat drept criteriu de clasificare va putea fi plasat într-un nod al unui arbore ce descrie clasificarea, din el pornind arce etichetate prin atomi pozitivi, negativi şi „neutri".

Metoda Davis-Putnam poate fi privită ca o metodă de analiză şi de reducere a unei probleme mai complexe la două probleme mai simple şi acestea reductibile, la rândul lor, la alte probleme. Ea cercetează ce se întâmplă dacă dăm prin ipoteză o valoare de adevăr unei variabile propoziționale sau unui atom predicativ. Ea poate fi privită şi ca o tehnică de simplificare dihotomică, dă în mod alternativ unui literal valoarea 1 şi apoi valoarea 0, dacă eşuăm în prima alternativă de a găsi un model care satisface formula complexă inițială. Valorizarea unei variabile ℓ prin 1 ne conduce la $NEG_\ell(S)$, iar valorizarea acesteia prin 0, ne conduce la $POS_\ell(S)$.

Pe baza notațiilor introduse şi a definițiilor D1-D6 putem lesne demonstra teoremele:

T4 $S \supset ((\ell = 1) \supset NEG_\ell(S))$

1) S	ip.
2) $\ell = 1$	ip.
3) $(\ell \vee B_i) \wedge (-\ell \vee C_j) \wedge R$	RE, 1), D6
4) $\ell \vee B_i$	E$\wedge$, 3
5) $-\ell \vee C_j$	E$\wedge$, 3
6) R	E$\wedge$, 3
7) $-\ell = 0$	neg -, 2)
8) $0 \vee C_j$	RE, 5), 7)
9) C_j	$0 \vee A = A$, 8)
10) $C_j \wedge R$	I$\wedge$, 9), 6)
11) $NEG_\ell(S)$)	RE, 10), D5

T5 $S \supset ((\ell = 0) \supset POS_\ell(S))$

T5 se demonstrează în mod analog cu T4.

Teoremele T5 şi T4 exprimă cele două alternative de a exclude un literal ℓ dintr-un set de clauze S: a-i da valoarea 1 şi a obține astfel $NEG_\ell(S)$ sau a-i da valoarea 0 şi a obține astfel $POS_\ell(S)$.

3. Teorema scindării şi principiul rezoluției

Suntem îndreptățiți să ne întrebăm ce legătură există între teorema scindării şi principiul rezoluției şi corespunzător, între metoda Davis-Putnam şi metoda rezoluției.

Vom observa, mai întâi, că atât teorema scindării, cât şi principiul rezoluției sunt scheme valide de inferență întemeiate pe utilizarea unor seturi de clauze disjunctive.

În al doilea rând, ambele conduc la eliminarea unei perechi opuse de literali, $<\ell, -\ell>$, care apar în două sau mai multe clauze anterioare. O aplicarea unică a principiului rezoluției se încheie prin identificarea unei consecințe imediate, în timp ce o aplicare unică a teoremei scindării reduce problema realizabilității unui set inițial de clauze la realizabilitatea a două seturi alternative de clauze diferite între ele prin modul de valorizare a literalului ℓ.

În al treilea rând, principiul rezoluției a fost creat de către J. A. Robinson pentru mecanizarea actelor inferențiale și pentru executarea unor demonstrații automatizate, în timp ce teorema scindării a fost creată în cercetări de semantică logică, de teoria modelelor. În ordinea intereselor cognitive urmărite de către Robinson, importante erau lanțurile de inferențe succesive, derivarea și respingerea rezolutivă, două procese dinamice prin care putem descoperi consecințe logice, adevăruri implicite ascunse într-un set inițial de date sau într-o bază de cunoștințe. Pentru creatorii teoremei scindării obiectivul l-a constituit permanent găsirea unui model pentru o bază de cunoștințe redată sub forma standard a unui set de clauze disjunctive sau, corespunzător, sub forma normală conjunctivă.

Prin interesul ei pentru scindare în funcție de un atom sau literal luat drept criteriu, cercetarea întreprinsă de către Davis și H. Putnam este relevantă pentru problematica teoriei clasificării, în timp ce, în cercetările lui Robinson, accentul cade pe teoria deducției.

Asemănările și deosebirile dintre cele două direcții de cercetare pot fi evidențiate de definirea conceptului de mulțime a rezolvenților imediați dintr-un set de clauze S printr-un literal dat ℓ din alfabetul A al lui S sau a „rezoluției majuscule", notată prin $RES_\ell(S)$.

D7. $RES_\ell(S) = S_\ell^0 \cup \{res_\ell(k_1, k_2) : k_1 \in S_\ell^+, k_2 \in S_\ell^-\}$

Acest concept intervine în teorema RES majuscul.

T7. Teorema RES majuscul. Mulțimea clauzelor neutre în raport cu un literal dat ℓ dintr-un set de clauze S reunită cu mulțimea rezolvenților imediați din S în același literal este echivalentă cu reuniunea dintre $POS_\ell(S)$ și $NEG_\ell(S)$. Simbolic:

$RES_\ell(S) \equiv POS_\ell(S) \cup NEG_\ell(S)$.

Demonstrație.

1) $RES_\ell(S)$ ip.
2) $S_\ell^0 \cup \{res_\ell(k_1, k_2) : k_1 \in S_\ell^+, k_2 \in S_\ell^-, \}$ 1), RE, D7
3) $S_\ell^0 \cup k_1/\{\ell\} \cup k_2/\{-\ell\}: k_1 \in S_\ell^+, k_2 \in S)\}$ 2),RE, (I),16.1
4) $S_\ell^0 \cup k_1/\{\ell\}: k_1 \in S_\ell^+ \cup S_\ell^0 \cup k_2/\{-\ell\}: k_2 \in S\}$ 3), TM
5) $POS_\ell(S) \cup NEG_\ell(S), \forall \ell(\ell \in A)\}$ 4), RE, D2.4,D2.5, 19.1

Întrucât în demonstrație s-au folosit numai regula substituirii echivalentelor , RE, și echivalențe, derivarea este valabilă și de la 5) la 1). În consecință, nu mai este necesară o demonstrație a implicației inverse.

Setul de clauze RES_ℓ (S) nu conține, în mod explicit, clauzele părinți din care au fost derivați rezolvenții imediați, dar aceștia pot fi obținuți, la nevoie, prin scheme de inferență ale algebrei booleene de forma $A = A \vee (C \wedge \text{-}C)$, unde A stă pentru o clauză rezolvent imediat.

Conceptul RES_ℓ (S) poate fi generalizat sau rafinat în mai multe sensuri.

Definiția D7 și teorema RES majuscul presupune selectat un literal ℓ dintr-un set de clauze S, în raport cu care sunt calculați toți rezolvenții imediați sau în raport cu care sunt calculate mulțimile de clauze POS_ℓ (S) și NEG_ℓ (S).. Putem să ne imaginăm situația în care dintr-un set de clauze S ne propunem să calculăm toți rezolvenții imediați, nu doar în raport cu un literal ℓ, ci în *raport cu toți literalii* care apar în S. Mulțimea literalilor din S provine din alfabetul de variabile propoziționale ce apar în setul de clauze S, pe care convenim să-l notăm prin A. De data aceasta vom obține un concept *RES majuscul generalizat,* pe care convenim să-l notăm prin RESG(S).

D8. $RESG(S) = \cup\{S_\ell^0 \cup \{res_\ell(k_1, k_2) : k_1 \in S_\ell^+, k_2 \in S_\ell\} \forall \ell(\ell \in A)\}$

Fie S un set de clauze și $A = \{l_1, l_2, \ldots l_n\}$ mulțimea literalilor care apar în S iar B o submulțime din A, i.e. $B \subset A$. Definim, atunci mulțimea de rezolvenți sau consecințe imediate din S în literali din submulțimea B:

D8. $RES_B(S) = \cup RES_\ell(S) : \ell \in B, B \subset A, \forall \ell(\ell \in B)$.

$RES_B(S)$ va cuprinde, deci, mulțimea clauzelor „neutre" în raport cu toți literalii din B și a tuturor clauzelor rezolvenți imediați din S, într-un literal din submulțimea B din A , unde A este mulțimea literalilor care apar în S.

$RES_B(S)$ poate fi redat analitic sub forma:

D8' $RES_B(S) = S_{\ell_i}^0 \cup res_{\ell_i}(k_1, k_2)$ $k_1 \in S_{\ell_i}^+, k_2 \in S_{\ell_i}^-, \forall \ell_1 \in B$, $1 \le i \le m, m < n$

Este evident că $RES_B(S)$ este o submulțime a lui RESG(S) .

Fiind dată o bază de cunoștințe transpusă clauzal, B poate fi alcătuit dintr-o listă de atomi predicativi relevanți pentru rezolvarea unei probleme; elementele lui B pot intra în datele problemei.

T8. $RES_B(S) \subseteq S \cup \{res_{\ell_i}(k_1, k_2) : k_1 \in S_{\ell_i}^+, k_2 \in S_{\ell_i}^-,$
$\forall \ell_1(\ell_1 \in B, 1 \le i \le n)\}$

Demonstrație.

1) $RES_B(S)$ ip

2) $S_{\ell_i}^0 \cup res_{\ell_i}(k_1, k_2) : k_1 \in S_{\ell_i}^+, k_2 \in S_{\ell_i}^-, \forall \ell_i (\ell_i \in B), 1 \leq i \leq n, m < n$, RE, 1), D8'

Dar $B \subset A$, căci B este un subalfabet al alfabetului A în care sunt scrise clauzele din S. Fie m cardinalul lui B. Întrucât $B \subset A$, vom avea $m < n$, unde n este cardinalul lui A sau numărul literelor (variabilelor) din alfabetul lui A.

Să considerăm acum mulțimea elementelor lui S. Aceasta poate fi scindată într-o submulțime de perechi irezolubile prin literalii din B sau „neutră" în raport cu variabilele incluse în B și într-o submulțime de perechi rezolubilă prin literali din B.

Fie k_1 și k_2 o pereche din S rezolubilă într-un literal ℓ din B. Atunci rezolventul ei va fi adevărat, potrivit teoremei rezolventului:

3) $res_\ell (k_1, k_2)$

Din 3) pe baza unei legi logice de algebră booleană, $(A \vee (C \wedge \text{-}C)$, putem scrie

4) $res_\ell (k_1, k_2) \vee (\ell \wedge \text{-}\ell)$

de unde putem reconstitui cele două clauze părinți din S sub forma:

5) $k_1 = res_\ell (k_1, k_2) \vee \ell$

6) $k_2 = res_\ell (k_1, k_2) \vee \text{-}\ell$

Fie, acum, un literal ℓ din B pentru care nu există în S vreo pereche de clauze rezolubile în ℓ. În consecință, orice clauză din S va fi „neutră" în raport cu literalul ℓ, aparținând submulțimii $S_{\ell_i}^0$ din 2).

Cu aceasta, demonstrația se încheie.

T9. $RES_B(S) \subset RESG(S)$

Demonstrație.

Admitem, ca și mai sus, că $B \subset A$, respectiv că B este un subalfabet al lui A și că dacă $|B| = m$ și $|A| = n$, atunci $m < n$:

1) $B \subset A$ ip.

2) $|B| = m$ ip.

3) $|A| = n$ ip.

4) $[(B \subset A) \wedge (|B| = m) \wedge (|A| = n)] \supset m < n$ TM

5) $m < n$ MP, 4), 1), 2),3)

6) $RES_B(S)$ ip.

7) $\cup\{S_\ell^0 \cup \{res_\ell (k_1, k_2) \quad k_1 \in S_\ell^+, k_2 \in S_\ell^-\}, \forall \ell(\ell \in B \wedge B \subset A)$, RE, 6),D8

Tot, potrivit lui D8, vom avem pentru $RES_A(S)$:

8). $\cup\{S_\ell^0 \cup \{res_\ell (k_1, k_2) : k_1 \in S_\ell^+, k_2 \in S_\ell^-\}\}, \forall \ell(\ell \in A)$, RE, D8

9) $RES_B(S) \subset RESG(S)$, TM, 8), 9)

Putem încerca o extindere a noțiunii de „neutralitate" prin trecerea de la o raportare la un element din A la o submulțime de elemente, B, din A.

D9. $S_B^0 = \{k \in S: \exists \ell(\ell \in B \wedge B \subset A \wedge alf(S, A) \wedge \ell \notin k \wedge -\ell \notin k\}$, unde prin alf(S, A) se înțelege „ A este alfabetul setului de clauze S"

S_B^0 desemnează un set de clauze „neutre" în raport cu variabilele sau atomii din alfabetul B, respectiv clauze din S ce nu conțin literali din subalfabetul B.

Cu alte cuvinte, S_B^0 desemnează mulțimea clauzelor din S care nu conțin apariții ale unor elemente din subalfabetul B sau mulțimea tuturor clauzelor „neutre" în raport cu vreun element din subalfabetul B.

Conceptul extins de neutralitate va fi folosit pentru definirea noțiunii de închidere tranzitivă a principiului rezoluției.

Este ușor de observat că, în toate variantele de rezoluție extinsă definite mai sus (D7, D8, D8'), am inclus în definitor, sub o formă sau alta, mulțimea clauzelor „neutre" în raport cu un literal sau cu o mulțime de literali. Noi putem opera, însă și cu un concept care să lase de o parte mulțimea clauzelor „neutre" Acesta va fi un sens mai restrictiv de rezoluție majoră.

D10. $R_A(S) = \{res_\ell(k_1, k_2) : k_1, k_2 \in S, \forall \ell(\ell \in A)\}$

Setul de clauze $R_A(S)$ mulțimea tuturor rezolvenților imediați dintr-un set de clauze S, indiferent de literalul ℓ din A în care are loc rezoluția. În termeni comuni de logică clasică am putea spune mulțimea tuturor consecințelor logice imediate degajabile dintr-un set de premise S. Este ușor de demonstrat că:

T10. $R_A(S) \subset RESG(S)$

Demonstrația se poate face prin ipoteză, pe baza definițiilor D8 și D10.

Este important de observat că toate definițiile date mai sus rezoluției extinse descriu acte inferențiale imediate, scurte, de un singur pas, un fel de explorare a modalităților de start și nu excursii mentale de lungimi mai mari. Am putea să le numim pe acestea drept „explorări în lățime" (A se reciti în acest sens despre strategia explorării în „lățime").

Putem, evident, cerceta secvențele de acte rezolutive sau secvențele de acte inferențiale mediate de regulile metodei Davis Putnam sau închiderea tranzitivă a relației de derivare rezolutivă.

D11. $R_A^+(S) = R_A(S) \cup R_A(S \cup R_A(S)) \cup R_A(S \cup R_A(S) \cup R_A(S \cup R_A(S))) \cup$

Este uşor de observat că:
$R^1_A(S) = R_A(S)$;
$R^2_A(S) = R_A(S \cup R_A(S))$;
$R^3_A(S) = R_A(S \cup R_A(S) \cup R_A(S \cup R_A(S)))$

D12. $R^{n+1}_A(S) = R_A(S \cup R^1_A(S) \cup R^2_A(S) \cup R^3_A(S) \cup . \cup R^n_A(S))$.

Pentru a defini relaţia de închidere reflexivă şi tranzitivă a derivării rezolutive $R_A(S)$ ne sprijinim pe setul iniţial de clauze şi pe D11.

D13. $R^*_A(S) = S \cup R_A^+(S)$

$R^*_A(S)$ cuprinde, aşadar, totalitatea clauzelor iniţiale sau derivate şi descrie domeniul de referinţă al strategiilor rezolutive.

Teorema Davis Putnam întemeiază posibilitatea eliminării unui literal dintr-un set de clauze dat şi reducerea unei probleme mai complexe la două probleme mai simple. Ea poate fi văzută şi ca o evaluare parţială din perspectiva satisfacerii unei condiţii exprimate printr-o variabilă propoziţională sau a unui atom predicativ. Orice pas în aplicarea metodei Davis-Putnam poate fi văzut ca o completare a unei interpretări I printr-un nou descriptor, ca un moment sau o soluţie parţială în rezolvarea unei probleme logice mai complexe.

Pe măsură ce înaintăm în construcţia modelului atribuind valori de adevăr unor variabile sau atomi predicativi eliminăm clauze sau literali din clauze, simplificăm setul iniţial de clauze, ajungem la dispariţia acestuia, dacă acest fapt este realizabil sau la obţinerea unei contradicţii explicite, dacă acest fapt este irealizabil.

În metoda Davis Putnam, ca şi în metoda interpretărilor semantice succesive intervin *valorizări impuse* de ipoteza iniţială, după care setul iniţial ar avea cel puţin o interpretare care l-ar putea face adevărat sau valorizări *asumate liber* de către decident. Valorizările impuse decurg din regula clauzei unitare, din regula literalului pur sau din regula clauzei consecinţe. Valorizările asumate liber provin din aplicarea regulii scindării. Depinde de decident, dacă acesta calculează mai întâi $POS_l(S)$ sau $NEG_l(S)$.

Atât metoda rezoluţiei, cât şi metoda Davis-Putnam, sunt computerizabile.

Metoda rezoluţiei este mai mult tratată în literatura de informatică decât metoda Davis-Putnam.

Enumerăm mai jos câţiva paşi în direcţia scrierii unui program formal pentru metoda Davis-Putnam.

1. Scrie setul S de clauze disjunctive sub forma unei mulţimi de mulţimi sau a unei liste de liste, pentru un eventual program în Prolog;

2. Scrie lista variabilelor sau atomilor predicativi care apar în setul iniţial de clauze;

3. Verifică dacă setul S conţine vre-o clauză unitară. Dacă S conţine o clauză unitară, atunci atribuie-i o valoare de adevăr care o face adevărată şi trece-o în interpretarea I;

4. Calculează, utilizând algebra logică, consecinţele valorizării clauzei unitare asupra celorlalte clauze ce o conţin şi asupra literalilor opuşi din clauzele ce conţin negaţia clauzei unitare(Aici se vor şterge unele clauze şi se vor elimina unii literali)

5. Dacă S se reduce la mulţiinea vidă de clauze, atunci S este realizabil; dacă S se reduce la clauza vidă, i.e. la o contradicţie, atunci S este irealizabil;

6. Dacă S este realizabil, atunci lista obţinută prin aplicarea instrucţiunii 3 ne va furniza o interpretare sau un model în care S devine adevărat.

4. Formele normale disjunctive perfecte şi metoda Davis-Putnam duală

În mod obişnuit, metoda Davis-Putnam se aplică la testarea realizabilităţii unor seturi de clauze disjunctive, i.e. la testarea termenilor unei forme normale conjunctive. În capitolul de faţă ne propunem să adaptăm metoda Davis-Putnam la testarea clauzelor conjunctive, respectiv la verificarea infirmabilităţii sau validităţii termenilor unei forme normale disjunctive.

4.1. **Noţiuni preliminare.** După cum am văzut, o formă normală disjunctivă este o disjuncţie de conjuncţii elementare sau o disjuncţie de conjuncţii de literali sau variabile propoziţionale, cu sau fără semnul negaţiei în faţă.

Orice formulă bine formată din logica propoziţiilor poate fi adusă la o formă echivalentă cu ea având forma unei disjuncţii de conjuncţii elementare, printr-un număr finit de transformări echivalente. Cum se realizează aceste transformări am văzut într-un capitol anterior (vezi cap. 5 şi urm. sau C. Popa, *Logica predicatelor*, Editura Hyperion, 1992, pp 95-96).

O formă normală disjunctivă a unei formule α, FND_α, are structura:

$$FND_\alpha = \vee k_j,\ j \geq 0 \qquad (1.1)$$

şi unde:

$$k_j = \wedge \ell_{ij},\ i \geq 0 \qquad (1.2)$$

În (1.1) şi (1.2) prin k_j (atenţie j) am notat clauze conjunctive sau conjuncţii de literali ℓ_{ij} sau variabile propoziţionale, cu sau fără semnul negaţiei în faţă.

De exemplu, forma normală disjunctivă a formulei:

$$\Psi = (p \supset (q \supset r)) \supset ((p \supset q) \supset (p \supset r)) \qquad (2)$$

$$FND_\Psi = pq\text{-}r \vee p\text{-}q \vee \text{-}p \vee r \qquad (3)$$
$$1 \quad 2 \quad 3 \quad 4$$

(Am convenit să nu mai scriem semnul conjuncţiei $\wedge$ în conjuncţiile elementare. Astfel, conjuncţia elementară $p \wedge q \wedge \text{-}r$ se va scrie pq-r)

Am observat în capitolul anterior că mulţimile de clauze, $POS_\ell(S)$ şi $NEG_\ell(S)$, atât în interpretarea clasică a teoremei Davis-Putnam, cât şi în interpretarea capitolului de faţă, provin dintr-o valorizare parţială, din atribuirea unei valori de adevăr unei variabile (sau unui atom predicativ) ce apare în forma normală.

Să admitem, de exemplu, că variabila p din (3) ia valoarea 1. Atunci (3) devine:

$$1q\text{-}r \vee 1\text{-}q \vee \text{-}1 \vee r \qquad (4)$$

Aplicând la (4) legile algebrei propoziţiilor:

$$1 \wedge A \equiv A \qquad (5)$$

$$\text{-}1 \equiv 0 \qquad (6)$$

rezultă:

$$q\text{-}r \vee \text{-}q \vee 0 \vee r \qquad (7)$$

de unde prin

$$0 \vee A \equiv A \qquad (8)$$

obţinem:

$$q\text{-}r \vee \text{-}q \vee r \qquad (9)$$

care prin:

$$A \vee (B \vee C) \equiv (A \vee B) \wedge (A \vee C) \qquad (10)$$

produce:

$$(\text{-}q \vee r \vee \text{-}r) \wedge (\text{-}q \vee r \vee q) \qquad (11)$$

care prelucrat, mai departe, pe baza legilor :

$$A \vee \text{-}A \equiv 1 \qquad (12)$$

$$1 \vee B \equiv 1 \qquad (13)$$

$$A \vee B \equiv B \vee A \qquad (14)$$

duce la relevarea caracterului tautologic al formulei (3).

Cercetăm valorizarea alternativă a formulei (3), p = 0.
Prin substituirea în (3) a lui p prin 0, obţinem:

$0q\text{-}r \vee 0\text{-}q \vee \text{-}0 \vee r$ (15)

Din (15) prin legile algebrei propoziţiilor:

$0 \wedge A \equiv 0$ (16)

$\text{-}0 \equiv 1$ (17)

$1 \vee B \equiv 1$ (18)

Rezultă, din nou că (3) este o tautologie.
Într-adevăr, (15) duce la:

$0 \vee 0 \vee 1 \vee r$ (19)

$1 \vee r$ (20)

1 (21)

Cum valorizarea are loc într-o logică bivalentă şi, deci, p nu poate lua decât valorile 0 sau 1 şi cum ambele valorizări au condus la concluzia că (3) este o lege logică, va trebui să acceptăm această concluzie.

Rezultatele finale ale celor două valorizări alternative sunt formulele (9) pentru p =1 şi (21) pentru p = 0. După cum vom vedea mai departe , acestea vor corespunde mulţimii de clauze $POS_\ell(S)$ şi respectiv $NEG_\ell(S)$, unde S este identic cu formula (3).

Trecerea de la (3) la (9) poate fi efectuată pe baza regulii:

Ra. Dacă S este o formă normală disjunctivă ce conţine literalul ℓ (în exemplul nostru p) şi dacă ℓ = 1, atunci elimină toate conjuncţiile elementare ce-l conţin pe $\text{-}\ell$ şi taie-l pe ℓ din conjuncţiile în care acesta apare pozitiv. S va fi realizabilă, dacă formula obţinută prin instrucţiunile de mai sus va fi realizabilă.

Dar instrucţiunile de mai sus revin la a păstra din S clauzele „neutre" în raport cu ℓ, respectiv conjuncţiile elementare care nu-l conţin nici pe ℓ, nici pe $\text{-}\ell$ şi la a păstra, de asemenea , conjuncţiile elementare care îl conţin pe ℓ, după ce omitem din ele pe ℓ. Or aceste exigenţe sunt notele definitorii ale setului de clauze $POS_\ell(S)$.

În mod analog, trecerea de la (3) la (20) satisface cerinţele obţinerii setului de clauze $NEG_\ell(S)$.

În fond, în ce a constat trecerea de la (3) la (20)?. Mai întâi, în omiterea conjuncţiilor elementare ce-l conţineau pe p, deoarece p era fals şi conjugarea cu falsul a unei formule ducea tot la fals iar falsul într-o disjuncţie duce la valoarea de adevăr a restului disjuncţiei . În consecinţă se şterg conjuncţiile elementare ce conţin o variabilă falsă.

Mai departe s-au aplicat regulile negaţiei, la nivelul constantelor logice 0 şi 1 (vezi, de exemplu (17)).

5. Teorema Davis-Putnam duală

Convenim să exprimăm schematic o formă normală disjunctivă în funcţie de un literal de referinţă ℓ prin echivalenţa:

$$S \equiv (\ell \wedge A) \vee (-\ell \wedge B) \vee C \qquad (\#)$$

unde A, B, C sunt metavariabile desemnând conjuncţii elementare în care nu apar literalii ℓ sau $-\ell$. Este lesne de observat că:

$$S_\ell^+ = \ell \wedge A \qquad (23)$$

$$S_\ell = -\ell \wedge B \qquad (24)$$

$$S_\ell^0 = C \qquad (25)$$

Pe baza definiţiilor de mai sus introducem:

$$POS_\ell(S) = A \vee C \qquad (26)$$

$$NEG_\ell(S) = B \vee C \qquad (27)$$

Formulele(23), (24), şi (25) descriu o partiţie a setului de termeni ai unei forme normale disjunctive în termeni ce conţin literalul de referinţă ℓ, S_ℓ^+, în termeni ce conţin negaţia acestuia, S_ℓ, şi în termeni ce nu nu-i conţin nici pe ℓ, nici pe $-\ell$, S_ℓ^0 $POS_\ell(S)$ provine din S atribuindu-i lui ℓ valoarea 1, iar $NEG_\ell(S)$ provine din S când îi atribuim lui ℓ valoarea 0.

Teorema Davis-Putnam duală. Fie S o formă normală disjunctivă sau un set de clauze conjunctive şi ℓ un literal din alfabetul A al lui S. Atunci S este realizabil, dacă şi numai dacă, $POS_\ell(S)$ este realizabil sau $NEG_\ell(S)$ este realizabil.

Demonstraţie.

1) S	ip.
2) $(\ell \wedge A) \vee (-\ell \wedge B) \vee C$	RE, 1), (#),
3) $\ell = 1$	ip. supl.1
4) $(\ell \wedge A) \vee (-\ell \wedge B) \vee C$	RS, 2), 3)
5) $A \vee (0 \wedge B) \vee C$	RE, 4),(5),(6)
6) $A \vee 0 \vee C$	RE, 5), (16)
7) $A \vee C$	RE, 6), (8)
8) $POS_\ell(S)$	RE, 7), (26)
9) $\ell = 0$	ip. supl.2
10) $(0 \wedge A) \vee (-0 \wedge B) \vee C$	RS, 2), 9)
11) $0 \vee (1 \wedge B) \vee C$	RE,10),(5),(6)

12) $(1 \wedge B) \vee C$ RE, 11),(8)
13) $B \vee C$ RE, 12), (5)
14) $NEG_{\ell}(S)$ RE,13), (27)

Pentru orice set S de conjuncţii elementare sau termeni ai unei forme normale disjunctive şi orice literal ℓ din alfabetul lui S , setul S este realizabil, dacă şi numai dacă este realizabilă cel puţin una dintre mulţimile de clauze A, B, C definite mai sus:

A este mulţimea conjuncţiilor de literali ce-l conţin pe ℓ , dar din care ℓ a fost tăiat ;

B este mulţimea conjuncţiilor elementare ce-l conţin pe $-\ell$, dar din care $-\ell$ a fost tăiat;

C este mulţimea de literali „neutri" în raport cu ℓ, respectiv mulţimea conjuncţiilor elementare de literali ce nu-l conţin nici pe ℓ, nici pe $-\ell$.

Teorema Davis-Putnam dual㠄 poate fi redată şi sub forma:

$S \equiv A \vee B \vee C$ (28)

unde A, B, C au sensul explicat mai sus.

6. Metoda Davis-Putnam duală

Dacă metoda Davis-Putnam se referă la seturi de clauze disjunctive sau la termenii formelor normale conjunctive, metoda pe care o propunem mai jos se referă la seturi de clauze conjunctive sau la termenii formelor normale disjunctive. Metoda creată de autorii americani este centrată pe ideea găsirii unei interpretări care să facă formula testată adevărată, duala ei este centrată pe ideea găsirii unui contramodel sau a unei interpretări care să facă formula testată falsă şi, deci, invalidă. Dimpotrivă, dacă găsirea unui contramodel eşuează în toate alternativele sistematic explorate, vom conchide că formula testată nu are contramodel şi, în consecinţă, este validă.

Vom formula regulile metodei Davis-Putnam duale dând prioritate regulii clauzei unitare, CU. Formulăm apoi regulile literalului pur, LP, regula clauzei consecinţă, CC, regula clauzei incompatibile şi, în sfârşit, regula scindării în noua sa formă.

Regula clauzei unitare, CU

Fie S un set de clauze conjunctive sau o formă normală disjunctivă şi ℓ un literal ce apare ca o clauză unitară în S. Atunci:

a) Atribuie-i lui ℓ o valoare care să-l facă fals (dacă ℓ apare pozitiv, atribuie-i valoare 0; dacă ℓ este un literal negativ, i.e. apare ca o clauză unitară $-\ell$, atunci atribuie-i lui ℓ valoarea 1);

b) Taie toate clauzele conjunctive ce conţin literalul clauză unitară, căci aceasta având valoarea 0 va face, inevitabil, false toate conjuncţiile în care apare;

c) Taie literalul opus clauzei unitare din conjuncţiile în care apare acesta, căci acesta va avea valoarea 1 şi $1 \wedge A \equiv A$

Setul iniţial S de clauze conjunctive va fi infirmabil, dacă setul obţinut prin instrucţiunile a), b), c) va fi infirmabil.

Exemplul 1

Fie S = p-r ∨ q-r ∨ -p-q ∨ r

Clauza unitară este r. Fiind un literal pozitiv şi dorind să facem clauza unitară infirmabilă valorizăm r = 0, i.e. I = { -r, }. Tăiem pe r, singura clauză în care apare r şi înlocuim pe -r din clauzele 1 şi 2 prin valoarea sa, 1. Astfel, obţinem:

1) p ∨ q ∨ -p-q

La formula 1) aplicăm, din nou, regula clauzei unitare în literalul p. Valorizăm p = 0, eliminăm pe p şi tăiem pe -p din clauza 3. Obţinem:

2) q ∨ -q

Expresia 2) este o tautologie şi ea nu este infirmabilă. Aplicând la 2), regula clauzei unitare, CU , îi dăm lui q valoarea 0, îl tăiem, iar negaţiei sale din clauza a doua îi dăm valoarea 1, care nu mai poate fi prelucrată prin regulile date.

Setul de clauze din exemplul 1 nu este infirmabil. În consecinţă el descrie o formulă validă.

Regula literalului pur, LP

Numim literal pur în S un literal al cărui opus nu apare în S.

Fie ℓ un literal pur care apare în S. Atunci:

a) Atribuie-i lui ℓ o valoare care să-l infirme;

b) Taie toate conjuncţiile elementare care au devenite false prin valorizarea efectuată.

S este infirmabil, dacă şi numai dacă setul de clauze S1, obţinut prin executarea instrucţiunilor a) şi b) este infirmabil.

Exemplul 2

S = p-q ∨ r-q ∨ -pq ∨ sr ∨ q-s ∨ r

Singurul literal pur în S este r. El poate fi infirmat prin valoarea r = 0. Eliminăm conjuncţiile elementare car-l conţin pe r şi care prin această valorizare devin false. Obţinem astfel:

S1 = p-q ∨ -pq ∨ q-s

În S1 apare literalul pur -s. Valorizând s = 1, q-s devine fals şi dispare. Obținem:

S2 = p-q ∨ -pq

S2 nu poate fi prelucrat prin regulile CU şi LP. Pentru prelucrarea lui va trebui să folosim regula scindării.

Regula clauzei contradictorii, CCon

Dacă S conține o conjuncție elementară de forma A-AB, aceasta se taie. S va fi infirmabil, dacă S1, obținut din S prin tăierea lui A-AB , va fi infirmabil.

Exemplul 3

S = pqr ∨ p-pq ∨ pr ∨ -pr ∨ -q-r

S1 = pqr ∨ pr ∨ -pr ∨ -q-r CCon

Regula clauzei consecință sau regula infirmării antecedentului, IA

Fie S un set de clauze conjunctive şi fie ABC şi A două clauze din S. Întrucât A este o consecintă din ABC şi A este infirmabilă pentru A = 0, pentru aceiaşi asignare de valori va fi infirmabilă şi clauza antecedent ABC.

Ca regulă de inferență putem scrie:

$$\frac{A = 0}{ABC = 0}$$

Exemplul 4

S = p-qr ∨ -q-s ∨ -ps ∨ pr

Întrucât p-qr ⊨ pr şi pr este infirmabilă pentru p = 0 sau q = 1, pentru aceiaşi atribuire va fi infirmabilă şi p-qr. Deci, oridecâteori se taie o formulă consecință ca infirmabilă se va tăia şi antecedentul ei. Este uşor de observat că această regulă este analoagă cu schema de inferență *modus tollens* din logica clasică.

Regula scindării, RS

Fie S un set de clauze conjunctive sau termenii unei FND la care nu se pot aplica regulile CU, LP, CCon, CC (sau IA). Atunci:

a) Alege un literal ℓ din S şi calculează $POS_\ell(S)$;

b) Verifică dacă $POS_\ell(S)$ este infirmabil. Dacă $POS_\ell(S)$ este infirmabil, atunci S este infirmabil;

c) Dacă $POS_\ell(S)$ nu este infirmabil, atunci calculează pe $NEG_\ell(S)$ şi verifică dacă $NEG_\ell(S)$ este infirmabil. Dacă $NEG_\ell(S)$ este infirmabil, atunci S este infirmabil;

d) Dacă nici $POS_\ell(S)$ nu este infirmabil şi nici $NEG_\ell(S)$ nu este infirmabil, atunci S este valid.

Exemplul 5

Fie formula:

$S = -pq \vee -q-s \vee p-t \vee t-s \vee s-v \vee v-u \vee su$

La setul S de clauze conjunctive nu se poate aplica niciuna dintre regulile CU, LP, IA, Ccon. Trebuie în mod necesar să aplicăm regula scindării, RS. Calculăm $POS_s(S)$ şi obţinem:

1) $-pq \vee -q \vee p-t \vee t \vee v-u$

Aplicăm regula CU_t la 1):

2) $-pq \vee -q \vee p \vee v-u$

Din nou CU în literalul p, i.e. CU_p şi obţinem:

3) $q \vee -q \vee v-u$

Dar 3) nu este infirmabil, deoarece conţine o pereche opusă de literali legaţi prin $\vee$ sau principiul terţiului exclus şi acesta este o lege logică. Vom conchide, deocamdată, că $POS_s(S)$ nu este infirmabil. Ne rămâne să verificăm infirmabilitatea lui $NEG_s(S)$. Calculăm, prin urmare, $NEG_s(S)$.

4) $-pq \vee p-t \vee -v \vee v-u \vee u$

Aplicăm la 4) regula CU_u şi obţinem:

5) $-pq \vee p-t \vee -v \vee v$

Dar 5) nu este infirmabilă pentru că 5) conţine $-v \vee v$. Prin urmare, $NEG_s(S)$ nu este infirmabilă. Deoarece, nici $POS_s(S)$, nici $NEG_s(S)$ nu sunt infirmabile, setul de clauze S nu este infirmabil. Prin urmare, setul de clauze S descrie o lege logică.

Exemplul 6

Se dă formula:

$\Phi = [(p \vee q) \wedge -r] \supset (q \supset s)$

să se decidă asupra infirmabilităţii ei prin metoda Davis-Putnam aplicată asupra FND.

Aducem formula Φ la FND:

1) $-[(p \vee q) \wedge -r] \vee (-q \vee s)$	RE , Φ, $A \supset B \equiv -A \vee B$
2) $[-(p \vee q) \vee r] \vee (-q \vee s)$	RE, De Morgan, 1)
3) $[(-p \wedge -q) \vee r] \vee (-q \vee s)$	RE, De Morgan, 2)
4) $(-p \wedge -q) \vee r \vee -q \vee s$	RE 3), asociat. $\vee$
5) $-p -q \vee r \vee -q \vee s$	Conv. scriere, 4)

Aplicăm, pe rând, regula CU în literalii, s, -q şi r şi obţinem mulţimea vidă de clauze, notată, în mod curent, prin ϕ (E o simplă coincidenţă că ϕ este minuscula lui Φ numele formulei testate !). Redăm, mai jos, etapele:

6) -p -q ∨ r ∨ -q CU_s , 5)
7) -p -q ∨ -q CU_r , 6)
8) ϕ CU_{-q} , 6)

Interpretarea ce infirmă formula iniţială este I = {q, -r -s,}, respectiv asignarea de valori: q= 1, r = 0, s = 0. Cu alte cuvinte contramodelul lui Φ este:

CM(Φ) = {q= 1, r = 0, s = 0}

Verificând aceste atribuiri de valori pentru formula iniţială Φ, vom obţine,

Φ = 0.
a) Φ = [(p ∨ 1) ∧ -0] ⊃ (1 ⊃ 0)
b) Φ = 1 ∧ 1 ⊃ 0
c) Φ = 1 ⊃ 0
d) Φ = 0.

Concluzii

Am demonstrat, în acest capitol, o teoremă duală teoremei Davis-Putnam, teoremă ce stă la baza unei metode de decizie analoage metodei Davis-Putnam clasice. Am formulat, totodată, regulile noii metode păstrând denumirile vechi, pentru a facilita transferul de deprinderi calculatorii dobândite în utilizarea metodei clasice.

Metoda Davis-Putnam duală este o metodă apagogică, deoarece ea, ca şi metodele arborilor de decizie sau tablourilor semantice, *caută sistematic un contramodel* şi dacă nu- l găseşte conchide coercitiv asupra validităţii formulei testate.

Regulile metodei Davis-Putnam, atât în versiunea lor clasică, cât şi în versiunea prezentată de noi mai sus, sunt scheme de inferenţă prin care reducem adevărul sau falsitatea unei formule mai complexe la adevărul sau falsitatea unor formule mai simple. Pe această cale reducem o problemă mai complicată la una sau două probleme mai simple.

La baza metodelor de tip Davis-Putnam stă ideea de *valorizare parţială* sau de atribuire de valori unei variabile propoziţionale sau unui atom predicativ, care sub raport epistemic are la bază emiterea ipotezei că o anumită propoziţie ar putea fi adevărată. Teorema şi regula scindării operează cu două ipoteze alternative, p =1 sau p = 0,

cum se întâmplă adesea în situaţiile acţionale caracterizate prin date incerte. Ipoteza p = 1 duce la setul de clauze $NEG_p(S)$. iar ipoteza p = 0 duce la setul de clauze $POS_p(S)$. Tindem să credem că metodele de tip Davis-Putnam s-ar putea dovedi *utile în studierea raţionamentelor plauzibile şi în teoria argumentării.*

Metoda Davis-Putnam dual㠄 merită să fie comparată cu metoda rezoluţiei duale, aşa cum am comparat mai sus metoda Davis Putnam clasică, cu metoda rezoluţiei de tip Robinson sau „clasice"

Ca şi metodele rezolutive, metodele de tip Davis-Putnam au la bază structuri logice standardizate, cum sunt formele normale şi sunt, în general, compatibile programele de demonstraţie automatizată.

Credem că metoda Davis-Putnam dual㠄 poate fi extinsă la limbajul logicii predicatelor de ordinul întâi. Literalii vor fi de data aceasta atomi predicativi deţinând variabile libere şi constante individuale. În acest caz, metoda va trebui să nuanţeze, în spiritul logicii predicatelor, conceptele de interpretare sau model şi să asimileze unele cerinţe ale algoritmilor de unificare. Dar despre aceasta vom vorbi într-un alt capitol.

Cap.9. FORMELE NORMALE, METODA REZOLUŢIEI ŞI ECUAŢIILE LOGICE

1. Formele normale şi rezolvarea de probleme

De regulă, ecuaţiile sunt egalităţi numerice în structura cărora intervin variabile sau necunoscute. În logica predicatelor vom descrie ecuaţiile ca egalităţi între termeni sau ca atomi predicativi de forma $t_1 = t_2$. Putem generaliza conceptul de ecuaţie astfel încât în locul numerelor să apară valori de adevăr din $\mathbf{B} = \{1, 0\}$. În acest caz o soluţie a unei ecuaţii logice va consta dintr-o atribuire de valori acordată variabilelor propoziţionale dintr-o formulă sau dintr-un set de formule care face formula sau întregul set de formule adevărată (adevărat). Grosier vorbind, soluţia va putea fi denumită *model* sau *interpretare ce satisface* o formulă sau un set de formule.

În logică facem o anumită licenţă poetică. În loc să scriem, ca în algebră, $p \wedge q = 1$, scriem doar $p \wedge q$ şi în loc de $p \wedge q = 0$, vom scrie $-(p \wedge q)$. Scrierea pozitivă a unei formule F echivalează în logică cu asertarea ei ca adevărată sau cu valorizarea ei cu 1, i.e. $F = 1$ sau $v(F) = 1$. În mod analog, scrierea ei negată echivalează cu asignarea ei ca falsă sau cu asertarea veridicităţii negaţiei ei.

Fiind dată o ecuaţie logică se poate pune problema descoperirii unei soluţii sau problema descoperirii tuturor soluţiilor care o satisfac.

Ecuaţiile logice pot avea una sau mai multe soluţii, pot fi inconsistente sau fără soluţii sau, dimpotrivă, pot fi tautologice, i.e. orice atribuire de valori să fie o soluţie a formulei sau sistemului de formule.

A rezolva o ecuaţie logică înseamnă a stabili, printr-un procedeu oarecare, dacă ecuaţia are soluţii sau nu şi a determina, în cazul că are, cel puţin o soluţie care o satisface.

În mod curent, pentru aflarea tuturor soluţiilor unei ecuaţii logice se utilizează metoda formelor normale, metoda matriceală sau metoda arborilor de decizie sau alte metode semantice adecvate. Dacă am adus formula sau setul de formule la FND orice termen al FND descrie o soluţie a ecuaţiei logice. Dar putem utiliza pentru aflarea

soluţiilor şi metoda interpretărilor semantice succesive sau metoda Davis-Putnam.

Presupunem dat un sistem de ecuaţii logice constând din formulele $F = F_1, F_2, \ldots F_n$ Pentru a afla soluţiile problemei prin metoda Davis-Putnam procedăm astfel:

1. Aducem fiecare formulă F_i cu $1 \le i \le n$, la forma ei normală conjunctivă, i.e. îi aflăm clauzele disjunctive;

2. Punem la un loc toate clauzele setului de formule iniţiale F şi obţinem setul de clauze disjunctive S_F ce corespund formulelor iniţiale F. Cum în logica propoziţiilor formele normale conservă valorile de adevăr ale formulelor iniţiale vom avea:

$v(F) = v(S_F)$ sau $F = S_F$

3. Testăm prin metoda Davis-Putnam setul de clauze S_F. Dacă prin aplicarea regulilor metodei Davis-Putnam obţinem din S_F mulţimea vidă de clauze Φ, atunci setul de formule iniţial F are o soluţie logică şi aceasta coincide cu interpretarea I obţinută prin aplicarea regulilor metodei Davis-Putnam(vezi cap.19).

Soluţiile unui set de formule pot fi obţinute şi prin metoda interpretărilor semantice succesive, dacă la punctul 3 de mai sus aplicăm la F_S metoda interpretărilor semantice succesive (vezi cap. 18).

Înainte de a prezenta rezolvarea sistemelor de ecuaţii logice prin metoda rezoluţiei pentru mai multă intuitivitate este util să evidenţiem nişte scheme de inferenţă aplicabile formelor normale. Să considerăm, pentru început, schemele de inferenţă aplicabile unei forme normale conjunctive, FNC, sau unui set de clauze disjunctive F_S.

$$\text{R1}\ \frac{v(F_S) = 1 \quad k \in F_S,\ \forall k}{v(k) = 1} \qquad \text{R2}\ \frac{v(k) = 1}{\exists \ell (\ell \in k \wedge v(\ell) = 1} \qquad \text{R3}\ \frac{1 \wedge k}{k} \qquad \text{R4}\ \frac{0 \wedge k}{0}$$

$$\text{R5}\ \frac{v(F_S) = 1 \quad F_S \models_{res} k}{v(k) = 1} \qquad \text{R6}\ \frac{F_S \models_{res} k \quad F_S \subset F'_S}{F'_S \models_{res} k} \qquad \text{R8}\ \frac{0 \vee A}{A}$$

R1.Dacă un set de clauze F_S este adevărat, atunci orice clauză k din F_S este adevărată.

R2. Dacă o clauză k este adevărată, atunci există în k un literal ℓ care este adevărat.

R3. Realizabilitatea conjuncţiei unei clauze k cu o formulă adevărată se reduce la realizabilitatea clauzei k.

R4. Conjuncţia unei clauze k cu o formulă falsă este falsă.

R5. Dacă dintr-un set adevărat de clauze F_S derivă rezolutiv clauza k, atunci aceasta este adevărată.

R6. Dacă dintr-un set adevărat de clauze F_S derivă rezolutiv clauza k şi dacă F_S este o submulţime a lui F'S, atunci din F'S derivă rezolutiv clauza k.

Din R1 şi R2 rezultă că un set de clauze F_S este adevărat dacă în fiecare clauză a sa există câte un literal adevărat.

$$R7\ \frac{v(F_S) = 1}{\forall k(k \in F_S \supset \exists \ell(\ell \in k \wedge v(\ell) = 1))}.$$

Altfel spus, setul de clauze F_S este adevărat, dacă putem găsi variabilelor propoziţionale care apar în F_S o atribuire de valori care să facă, în fiecare clauză, cel puţin câte un literal adevărat.

Problema pe care ne-o punem în acest moment este următoarea: Fiind dat un set de clauze F_S , care descrie o bază de cunoştinţe sau datele unei probleme, cum putem găsi, aplicând metoda rezoluţiei, soluţia acelei probleme sau o interpretare care satisface toate restricţiile problemei şi, în plus, să răspundă la câteva întrebări stipulate în textul problemei. Întrebările le putem caracteriza ca variabile propoziţionale sau predicate deschise, descrise prin formule bine formate, care devin propoziţii explicit adevărate, prin interpretarea I ce descrie soluţia problemei.

Admitem că F_S provine dintr-un set iniţial de formule iniţiale F, ce descrie formalizarea problemei într-un limbaj logic, în logica propoziţiilor sau logica predicatelor.

O întrebare într-o problemă ar putea fi descrisă prin formule de forma: ? p; ?(q $\wedge$-r) ; ?(s $\supset$t) ; ?R(x, y); ?$\exists$z(P(x,z) $\wedge$ Q(z, y)), etc.

Răspunsul la o întrebare sau o soluţie parţială a unei probleme va consta din derivarea din setul iniţial de clauze F_S, obţinut din datele problemei, a unei instanţieri a unui predicat sau a unei valorizări a unei variabile propoziţionale, variabilă care a apărut anterior prefixată de semnul întrebări. Răspunsul va fi, aşadar, o clauză derivată care transformă predicatul din propoziţia interogativă într-o aserţiune.

Întrucât formula iniţială F este echivalentă în logica propoziţiilor cu forma sa normală conjunctivă, F_S, vom avea:

$$v(F_S) = 1 \supset v(F) = 1$$

Cu alte cuvinte, orice interpretare ce satisface setul de clauze sau FNC va satisface şi setul de formule iniţiale ce descriu datele problemei. A găsi soluţia problemei înseamnă a găsi o interpretare

pentru F_S care să conducă, în acelaşi timp, la instanţierea predicatelor sau variabilelor propoziţionale care au apărut în textul întrebărilor. În logica propoziţiilor aceasta va însemna să derivăm ca teoremă literalul pozitiv care a fost prefixat în formalizarea întrebării de semnul'?'

Fie F un set de formule logice adevărate ce descriu datele unei probleme şi ?- p unica întrebare a problemei şi F_S forma normală conjunctivă a lui F sau setul ei de clauze. Atunci interpretarea $I = \{\ell_1, \ell_2, \ldots \ell_g\}$, cu $1 \leq g \leq m$, unde m este numărul de clauze din F_S, descrie o soluţie a problemei logice în cauză, dacă şi numai dacă. interpretarea I satisface deopotrivă pe F_S şi clauza p ce descrie răspunsul la întrebare.

Prin legile logicii propoziţiilor şi schemele de inferenţă definite mai sus vom avea:

1. $v(F) = 1$ ip.
2. $F_S \vdash_{res} p$ ip.
3. $v(F) \equiv v(F_S)$ LP
3. $v(F_S) = 1$ RE, 1, 2
4. $\forall k(k \in F_S \supset \exists \ell(\ell \in k \wedge v(\ell) = 1))$, R7, 4
5. $v(k) = 1, \forall k \in F_S$ R1, 4
6. $\exists \ell(\ell \in k \wedge v(\ell) = 1\}$ R3, 6
7. $\ell_i \in k \wedge v(\ell_i) = 1$ E $\exists$, 7
8. $\ell_i \in k$ $1 \leq i \leq n$ E$\wedge$,8
9. $v(\ell_i) = 1$, $1 \leq i \leq n$ E$\wedge$,8

Să admitem că în S sunt m clauze. Fiecare clauză este adevărată, căci F_S este o conjuncţie şi, potrivit lui 4, $v(F_S) = 1$, iar o conjuncţie nu poate fi adevărată fără ca fiecare termen al ei să fie adevărată. Scriem, deci:

11. $k_1 \wedge k_2 \wedge \ldots \wedge k_m$

Fiecare dintre clauzele k_j, cu $1 \leq j \leq m$, conţine, potrivit condiţiilor 7-10 de mai sus, câte un literal adevărat ℓ_i care garantează adevărul clauzei în care se află. Listând toţi aceşti literali obţinem o interpretare $I = \{\ell_1, \ell_2, \ldots \ell_g\}$, cu $1 \leq g \leq m$. Întrucât interpretarea I face adevărată fiecare clauză din F_S, putem scrie:

12. $\vdash_I F_S$

Din 12 şi ipoteza 2 rezultă, prin teorema derivării rezolutive (vezi şi schema de inferenţă R5) adevărul literalului p în interpretarea I:

13. $\vdash_I p$.

Interpretarea $I = \{\ell_1, \ell_2, \ldots \ell_g\}$, cu $1 \leq g \leq m$ este soluţia care satisface problema logică formulată mai sus.

Interpretarea I ce descrie soluția poate fi găsită cu ajutorul metodei rezoluției, dacă vom urmări obținerea ca rezolvenți din F_S a cât mai multe clauze unitare. În consecință, propunem următorul algoritm pentru identificarea soluției unui sistem logic alcătuit din setul de formule inițiale F, ce descriu datele problemei și din întrebarea ?p.

1. Se aduce F la forma normală conjunctivă, FNC sau la forma clauzală F_S ;

2. Se trec clauzele unitare din setul inițial de clauze F_S în interpretarea $I \models \{... \}$;

3. Derivăm din F_S toate clauzele unitare derivabile și le adăugăm la interpretarea I;

4. Asociem fiecărui literal obținut ca teoremă mulțimea clauzelor pe care acesta le face adevărate;

5. Dacă literalul interogație p este derivabil din setul inițial F_S și reuniunea mulțimilor de clauze făcute adevărate de către literalii derivați este echivalentă cu setul inițial F_S, atunci interpretarea I, obținută cum am precizat mai sus, descrie o soluție a problemei.

Se înțelege, apariția pozitivă a unei variabile, cum ar fi, de exemplu, p, va fi interpretată ca v(p) = 1, iar apariția negată a unei variabile , de exemplu, -q, va fi interpretată ca v(q) = 0 iar în interpretarea în curs de construire, ele vor fi scrise ca: I = {p, -q, . }.

Exemplul 1

$$F = ((q \supset p) \vee s) \wedge (p / q) \wedge (s \supset r) \wedge (q \supset t) \wedge (s \supset t) \wedge \text{-t}, \ ?(p \vee \text{-q})$$

$$F_S = \underset{1}{(\text{-q} \vee p \vee s)} \wedge \underset{2}{(\text{-p} \vee \text{-q})} \wedge \underset{3}{(\text{-s} \vee r)} \wedge \underset{4}{(\text{-q} \vee t)} \wedge \underset{5}{(\text{-s} \vee t)} \wedge \underset{6}{\text{-t}} \wedge \underset{7}{(p \vee \text{-q})}$$

Formula F_S de mai sus am obținut-o conform instrucțiunii 1 și este forma FNC a lui F, la care am adăugat clauza aflată sub semnul întrebării. Potrivit instrucțiunii 2, trecem clauza unitară 6 în interpretarea I în curs de construire și scriem I1 ={-t,. .}. Mai departe, prin rezoluție în t din clauzele 5 și 6 obținem rezolventul -s pe care îl trecem în interpretarea I1 și obținem I2 ={-t, -s,. }. La rândul lui, rezolventul -s, împreună cu clauza 1, produce clauza rezolvent -q ∨ p, ce coincide cu clauza aflată sub semnul interogației(vezi F, partea dreaptă) care, împreună cu clauza 2, rezolvă în literalul p ducând la clauza unitară -q, ce va trebui să fie trecută și ea în interpretarea în construcție, ajungându-se la interpretarea finală: I3 ={-t, -s, -q }.

Este ușor de observat că interpretarea I3 „acoperă" setul de clauze F_S :

-t garantează adevărul clauzei 6;

-s garantează adevărul clauzelor: 3, 5 ;

-q garantează adevărul clauzelor: 1, 2, 4.

Pentru problemele mai complexe, cu clauze mai numeroase și cu atribuiri de valori pentru mai multe variabile putem construi tabele de verificare a „acoperirilor" și de reprezentare a soluțiilor alternative.

Construim un astfel de tablou pentru setul de clauze F_S din exemplul 1.

ℓ / k	1	2	3	4	5	6	7
-t						*	
-s			*		*		
-q	*	*		*			*

Tabloul de verificare a adecvării soluției la datele problemei

Interpretarea I3 = { -t, -s, -q} satisface toate condițiile stipulate de datele problemei exprimate în datele problemei redate prin formula F și, respectiv, prin setul de clauze F_S, precum și clauza 7 ce exprimă răspunsul la întrebarea formulată.

Exemplul 2

Se dă setul de clauze:

$$S2 = \underset{1}{(-p \vee s)} \wedge \underset{2}{(p \vee -q \vee -t)} \wedge \underset{3}{(p \vee -r \vee -s)} \wedge \underset{4}{-p} \wedge \underset{5}{(-r \vee t)} \wedge \underset{6}{r}$$

și întrebarea 7.?(r/s) i.e. 7. $-r \vee -s$.

Să se găsească soluția ecuației logice S2, astfel încât să satisfacă 7.

Aplicăm schemele de inferență R1-R7 sau instrucțiunile algoritmului prezentat mai sus.

Instrucțiunea 1 este satisfăcută din start. Aplicăm instrucțiunea 2 și scriem:

I1 = {-p, r}.

Degajăm concluziile care rezultă prin legile negației și principiul consistenței în evaluare și eliminăm clauzele 1 și 6 și tăiem din clauzele 2, 3 și 5 literalii opuși celor trecuți în interpretarea I1 (vezi schemele de inferență R3, R8). Pe baza acestor operații obținem:

S2 a = (-q ∨ -t) ∧ -s ∧ t
2 3 5

Aplicând la S2a instrucţiunile 2 şi 3 scriem:

I2 = {-p, r, -s, t}

S2b = -q

de unde rezultă:

I3 = {-p, r, -s, t, -q}

Din I3 rezultă banal -p∨-q, argumentul întrebării iniţiale (vezi 7 mai sus).

Putem lesne construi tabloul de acoperire:

ℓ k	1	2	3	4	5	6	7
-p	*			*			*
r						*	
t					*		
-q		*					*
-s			*				

Tabloul de verificare a adecvării soluţiei la datele problemei din Exemplul 2

Soluţionarea ecuaţiilor logice prin metoda rezoluţiei presupune, desigur, aducerea în prealabil a formulelor date iniţial la forma normală conjunctivă sau la forma clauzală. Găsirea atribuirilor de valori ce satisfac formula se face treptat, prin obţinerea, rând pe rând, prin derivare rezolutivă, a unor clauze unitare ce satisfac un număr de clauze din setul iniţial S, până ce se ajunge la situaţia în care clauzele unitare obţinute acoperă întregul set de clauze iniţiale sau când se dovedeşte că nu există nici o atribuire care să acopere toate clauzele din S.

2. Clauzale disjunctive, arborii şi-sau şi instrucţiunile Prolog

Formele clauzale disjunctive se obţin prin aducerea unui set de formule la forma lor normală conjunctivă, care este o conjuncţie de disjuncţii elementare sau clauze disjunctive.

Trecerea de la un set de formule implicative ce descriu raporturi de dependenţă dintr-un anumit domeniu din realitate la forma lor normală conjunctivă înseamnă o reducere a formulelor iniţiale la o formulă unică

echivalentă cu acelea şi în acelaşi timp o formă de analiză a unor formule complexe în setul de clauze elementare ce le definesc.

Clauzele elementare obţinute pot fi de mai multe tipuri: 1. clauze unitare pozitive sau „fapte"; clauze unitare negative sau negaţia unor fapte; clauze negative sau clauze cu toţi literalii negaţi sau „întrebări"; clauze program care au exact un singur literal pozitiv, ceilalţi literali fiind toţi negativi; clauza vidă sau clauza fără nici un literal.

Demn de reţinut este faptul că în clauzele program literalii negaţi provin din antecedentul propoziţiei implicative, iar literalul pozitiv coincide cu literalul pe post de consecvent.

Ilustrăm aceasta printr-un exemplu simplu. Presupunem că în setul de formule implicative iniţiale aveam formula:

1. $(p \wedge q \wedge r) \supset s$	formulă condiţională iniţială
2. $-p \vee -q \vee -r \vee s$	clauza disjunctivă obţinută din 1 care este şi clauza program
3. s:- p, q, r.	regulă Prolog obţinută din 2 sau 1.

În formula 3, care este o instrucţiune sau regulă Prolog, s este „capul" instrucţiunii Prolog şi acesta coincide cu variabila sau literalul pozitiv pe post de consecvent în formula implicativă 1; simbolul „ :- " descrie implicaţia Prolog , iar formulele p, q, r descriu „coada" instrucţiunii Prolog. Virgula pusă între variabile ţine locul conjuncţiei.

Formula 3 se citeşte: „s este adevărat, dacă p şi q şi r sunt adevărate". Punctul de după r anunţă că scrierea instrucţiunii s-a terminat. Aşadar, o regulă Prolog descrie un raport de dependenţă condiţională citit de la consecvent sau consecinţă către antecedenţi sau condiţii. La rândul lor, termenii din antecedenţi p, q şi r, pot fi, fiecare dintre ei, adevăraţi, dacă alte condiţii sunt adevărate.

Acest demers repetat ne sugerează cum putem construi un arbore care să descrie raporturile de condiţionare într-o bază de cunoştinţe. În acelaşi timp, acest arbore citit de sus în jos ne va descrie demersul logic inferenţial care porneşte de la baza factuală plasată pe frunzele arborelului şi coboară, pas cu pas, prin regulile Prolog, spre rădăcina arborelui care descrie teza de demonstrat.

Să considerăm următoarea bază de cunoştinţe banală descrisă de propoziţiile:

Clauze generice sau reguli

1. Dacă deţinem instalaţia şi avem materia primă şi produsul se cere la piaţă, atunci îl producem.
2. Dacă am cumpărat instalaţia, atunci o deţinem.

3. Dacă sunt furnizori ai materiei prime şi avem bani, atunci avem materia primă.
4. Dacă produsul este de calitate şi este la un preţ convenabil, atunci se cere la piaţă.

Fapte

5. Am cumpărat instalaţia.
6. Sunt furnizori ai materiei prime.
7. Avem bani.
8. Produsul este de calitate.
9. Produsul este la un preţ convenabil.

Convenţii de notare

1. Am cumpărat instalaţia. = n
2. Sunt furnizori ai materiei prime. = v
3. Avem bani. = m
4. Produsul este de calitate. = w.
5. Produsul este la un preţ convenabil. = f.
6. Deţinem instalaţia . = p
7. Avem materia primă. = q
8. Produsul se cere pe piaţă. = r

Formalizăm în limbajul logicii propoziţiilor informaţiile din mica noastră bază de cunoştinţe. Totodată, vom scrie alături, în dreapta, instrucţiunile Prolog pentru aceste formule.

1. $(p \wedge q \wedge r) \supset s$	s:- p, q, r.
2. $n \supset p$	p :- n.
3. $(v \wedge m) \supset q$	q:- v, m.
4. $(w \wedge f) \supset r$	r:- w, f.
5. n.	n.
6. v.	v.
7. m.	m.
8. w.	w.
9. f.	f.

Aducem la o formă clauzală formulele 1-9.

$$B = \underset{1}{(-p \vee -q \vee -r \vee s)} \wedge \underset{2}{(-n \vee p)} \wedge \underset{3}{(-v \vee -m \vee q)} \wedge$$
$$\wedge \underset{4}{(-w \vee -f \vee r)} \wedge \underset{5}{n} \wedge \underset{6}{v} \wedge \underset{7}{m} \wedge \underset{8}{r} \wedge \underset{9}{f}$$

Să admitem că am trecut clauzele Prolog din dreapta formulelor 1-9 care transpun în limbajul logicii propoziţiilor propoziţiile în limba română 1-9 ca un nou program într-un limbaj de programare Arity sau Winprolog şi că punem întrebarea ?-s. Atunci sistemul va încerca să derive din setul de instrucţiuni 1-4 şi din clauzele factuale 5-9 predicatul s/0 . Dacă el reuşeşte să facă aceasta, atunci va răspunde: yes.

```
| ?- s.
yes
```

şi va aştepta o altă întrebare a noastră.

Principalele idei pe care vrem să le acreditäm în acest capitol sunt următoarele:

1.Cele mai multe enunţuri dintr-o limbă naturală, îndeosebi enunţurile care descriu raporturi de dependenţă, pot fi transpuse într-un limbaj logic cum este logica propoziţiilor sau limbajul logicii predicatelor de ordinul întâi. Acestea conţin, de regulă, implicaţii.

2. Formulele traduse într-un limbaj logic pot fi aduse la forme normale conjunctive sau clauze disjunctive care conservă relaţia de realizabilitate.

3. Clauzele disjunctive elementare reprezintă o analiză a condiţiilor stipulate de formulele de la punctul 1, care descriu o realitate fizică sau socială.

4. Dacă clauzele disjunctive elementare sunt clauze program, adică clauze ce conţin un singur literal pozitiv şi unul sau mai mulţi literali negativi, atunci acestea pot fi scrise ca reguli Prolog.

5. Dacă la regulile Prolog adăugăm datele factuale stipulate în baza de cunoştinţe sau propunem noi altele adecvate, atunci putem întreba programul Prolog despre o serie de predicate derivate, care apar în consecventul implicaţiilor Prolog sau în „capul" instrucţiunilor Prolog.

6. Demersul demonstrativ poate fi exprimat printr-un arbore de derivare sau printr-un arbore de întemeiere a unei teze în raport cu o bază de cunoştinţe dată sau presupusă de un agent epistemic

7. Pentru orice formă normală conjunctivă (sau pentru orice set de clauze disjunctive) putem construi, pornind de la un scop epistemic sau un predicat chestionat (?-r (X,Y). sau ?- p. sau altele de forma aceasta) un arbore de derivare.

Cum se construieşte un astfel de arbore de derivare ?

a. Plasăm în rădăcina arborelui predicatul sau propoziţia chestionată (de exemplu ?-r.; ?-s etc.).

b. Căutăm o clauză care să aibă drept „cap" predicatul chestionat (sau un fapt care să coincidă cu predicatul chestionat).

c. Predicatul chestionat sau „capul" regulei Prolog va fi adevărat, dacă va fi adevărată „coada" regulei Prolog.

d. Dacă literalii din coada clauzei depind, la rîndul lor, de adevărul altor literali, continuăm procesul regresiv, până când ajungem la literali care nu mai sunt capete ale altor clauze program. Aceştia vor fi numiţi noduri terminale sau „frunze"

e. Dacă fiecare „ frunză" corespunde unei clauze unitare sau unui fapt din baza de cunoştinţe B, atunci teza de demonstrat este pe deplin argumentată şi baza factuală şi regulile descrise în graf permit, descendent, prin *modus ponens* să conchidem asupra consecventului imediat şi de aici, tot aşa mai departe până la derivarea literalului sau propoziţiei existentă la rădăcina arborelui (în exemplul considerat s).

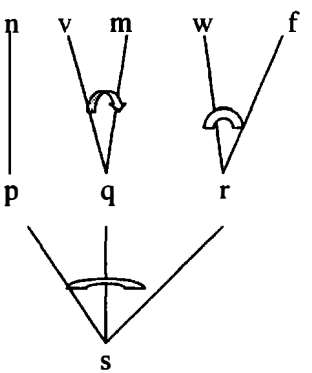


Fig. 1. Arbore de derivare

Fiecare unghi din Fig.1 descrie o regulă sau instrucţiune Prolog, căreia în setul de clauze disjunctive B îi corespunde o clauză program. Linia p-n descrie o clauză Prolog având „coada" sau antecedentul alcătuit dintr-un singur literal. Citit de jos în sus arborele descrie un demers de întemeiere sau argumentare. Citit de sus în jos descrie o inferenţă în *modus ponens* Prima instrucţiune Prolog aflată la baza arborelui ar putea fi citită: Producem (s), dacă deţinem instalaţia (p) şi avem materie primă (q) şi produsul se cere la piaţă (r). Apoi urmează argumentarea sau întemeierea premiselor lui s, respectiv a lui p, q şi

r, fiecare separat, respectiv a lui p prin n, a lui q prin v şi m şi a lui r prin w şi f.

Variabilele n, v, m, w şi f care stau pentru propoziţiile: „am cumpărat instalaţia" (n), „sunt furnizori de materie primă" (v) şi altele (vezi convenţiile de notare date mai sus) sunt plasate pe „frunzele" arborelui, căci ele nu pot fi reduse sau explicate prin alte reguli. Nu avem în programul Prolog dat mai sus nici o regulă care să aibă drept „cap" vreuna dintre literele n, v, m, w, şi f. Convenim să notăm mulţimea frunzelor prin Fr. La fel, putem conveni să notăm mulţimea clauzelor unitare pozitive sau a faptelor prin simbolul Fact. Toate clauzele marcate printr-un arc dublu descriu arce „şi". În Fig 1 nu apar arce SAU în care acelaşi simbol C, ce etichetează un nod oarecare să poată fi susţinut prin două mulţimi de premise distincte, respectiv două reguli Prolog să aibă acelaşi „cap"

Fr = [n, v, m, w, f].

Fact =[n, v, m, w, f].

Nu în toate cazurile mulţimea frunzelor coincide cu mulţimea faptelor sau clauzelor factuale.

Condiţia necesară pentru întemeierea materială a unei argumentări (aici a lui s) este ca mulţimea frunzelor sau a suporturilor ultime Fr pentru susţinerea tezei de argumentat (aici a lui s) să fie inclusă în mulţimea clauzelor factuale, Fact.

$Fr \subseteq Fact$

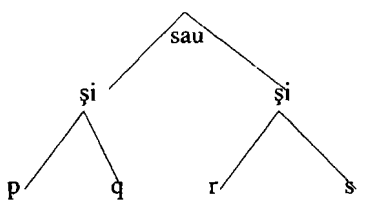
Dimpotrivă, dacă există cel puţin o „frunză" care nu e inclusă în mulţimea faptelor, atunci teza de argumentat din rădăcina arborelui eşuează. Tezele neincluse în mulţimea frunzelor vor trebui întemeiate pe alte premise incluse în mulţimea faptelor descrise evident de reguli Prolog suplimentare, care vor avea toate frunzele incluse în mulţimea clauzelor factuale.

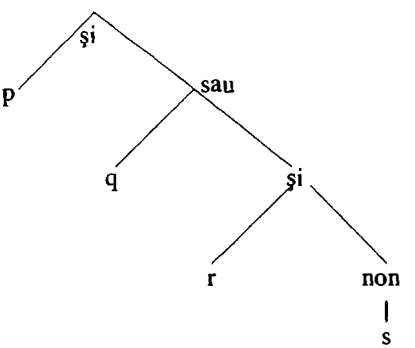
3. Logica propoziţiilor şi programarea logică

Am văzut anterior mai multe procedee de decizie „manuale" Prezentăm acum un procedeu de decizie automatizată a formulelor din logica propoziţiilor scris în Prolog. Admitem că variabilele propoziţionale desemnează propoziţii declarative din limba naturală sau din teoriile unor ştiinţe particulare şi că acestea pot fi adevărate sau false.

De exemplu, p = stau acasă; q = învăţ la analiză; r = mă duc la Sinaia, s = mă distrez ; etc.Optăm, de asemenea, pentru modul de scriere prefixială, după care scriem, mai întâi, conectivul logic şi după el, în ordine, argumentele sale. Conectivele logice vor fi redate ca în limba română prin „şi", „sau", „non" (celelalte fiind reductibile la acestea) şi nu prin simbolurile specifice: $\wedge$, $\vee$, - . O funcţie logică de forma: $(p \wedge q) \vee (r \wedge s)$ va fi scrisă ca: sau(şi(p, q), şi(r, s)), iar o funcţie de forma: $p \wedge (q \vee (r \wedge -s))$ va fi scrisă ca: şi(p, sau(q, şi(r, non(s)))).

Structurile unor astfel de formule pot fi redate neechivoc cu ajutorul unor arbori:





Programarea logică poate fi raportată în mod firesc la formele normale conjunctive şi la teoria mulţimilor de clauze, deoarece ea se întemeiază pe utilizarea clauzelor în scopuri calculatorii, pentru că ea modelează derivări rezolutive.

Deoarece faptele şi regulile Prolog sunt în formă clauzală, forma particulară a clauzelor Horn distincte, ele se mai numesc şi *clauze Prolog*.

Numim *clauza Horn* o clauză ce conţine cel mult un literal pozitiv. O clauză Horn poate fi:1. o clauză unitară pozitivă, formată dintr-un singur literal pozitiv (literal nenegat); 2.o clauză negativă, formată numai din literali negativi; 3 o clauza formată dintr-un literal pozitiv şi cel puţin un literal negativ, numită şi clauză Horn mixtă; 4 clauza vidă, notată prin

Se numeşte *clauză Horn distinctă* sau *clauză program* o clauză care are exact un singur literal pozitiv, ea putând fi, deopotrivă, o clauză unitară pozitivă, fie o clauză Horn mixtă.

Clauzele Horn unitare pozitive descriu în Prolog aserţiuni sau fapte, iar clauzele Horn mixte descriu propoziţii condiţionale sau reguli.

O clauză Horn mixtă provine, de regulă, dintr-un enunţ condiţional de forma:

$$(A_1 \wedge A_2 \wedge . . \ . \wedge A_n) \supset B$$

care prin regulile eliminării implicaţiei şi legile De Morgan devine:

$$-A_1 \vee -A_2 \vee ... \vee -A_n \vee B$$

şi se poate reda în Prolog prin instrucţiunea:

$$B :- A_1, A_2, ..., A_n .$$

Instrucţiunea Prolog porneşte de la propoziţia consecinţă spre propoziţia antecedent sau temei al raţionamentului. În Prolog scriem „mă îmbăt, dacă beau" şi nu „dacă beau, mă îmbăt". „Mă îmbăt" va fi pe post de cap al instrucţiunii Prolog (vezi mai sus B), iar „beau" va sta pe post de cap al instrucţiunii Prolog (vezi mai sus A_1 A_2. . , A_n), iar semnul implicaţiei logice de la consecinţă la antecedent va fi redat prin „:-".

Redăm mai jos, după F. Giannesini şi colaboratorii (Prolog, Intereditions, 1985) un set de instrucţiuni Prolog pentru decizia unor formule din logica propoziţiilor, scrise conform unei notaţii frontale, de tip polonez. Instrucţiunile pe care le scriem captează în fapt procedeul de decizie matriceal. Pentru scrierea instrucţiunilor vom folosi un predicat de forma: val(p,v) i.e. „Valoarea de adevăr a formulei p este v"

Într-o primă formă, cea mai intuitivă, dar nu şi cea mai economică, programul pentru decizia unei formule logic propoziţionale, cuprinde următoarele 15 instrucţiuni:

```
    R1 val(0, 0).
    R2 val(1, 1).
R3 val(si(p, q), b) :-
        val(p, u),
        val(q, v),
                comb_si(u, v, b).
    R4 val(sau(p, q), b):-
                val(p, u),
                val(q, v),
                comb_sau(u, v, b).
    R5 val(non(p), b):-
                val(p, u),
                comb_non(u, b).
    R6 comb_si(1, 1, 1).
    R7 comb_si(1, 0, 0).
```

R8 comb_si(0, 1, 0).
R9 comb_si(0, 0, 0).
R10 comb_sau(1, 1, 1).
R11 comb_sau(1, 0, 1).
R12 comb_sau(0, 1, 1).
R13 comb_sau(0, 0, 0).
R14 comb_non(0, 1).
R15 comb_non(1, 0).

R1 şi R2 sunt reguli de terminare a programului vizând formule „frunze", inanalizabile mai departe. Ele spun că valoarea de adevăr a formulei 0 este 0 şi că valoarea de adevăr a formulei 1 este 1; 0 şi 1 sunt deopotrivă formule şi valori de adevăr.

Formula R3 se citeşte: valoarea de adevăr a formulei si(p, q) este b, dacă valoarea de adevăr a formulei p este u şi valoarea de adevăr a formulei q este v şi combinând valorile de adevăr u şi v după regulile de combinare şi obţinem b.

În mod analog se citesc regulile R4 şiR5.

Regulile R6-R15 sunt, de fapt, transpuneri în instrucţiuni Prolog a definiţiilor matriciale ale operatorilor conjuncţie, disjuncţie şi negaţie.

Demn de reţinut este faptul că regulile R3, R4 şi R5 sunt reguli recursive, care se repetă de mai multe ori, până ce se ajunge la aplicarea regulilor R1 şi R2 ce funcţionează ca reguli de oprire a demersului calculatoriu.

Dacă dispunem de un calculator dotat cu un limbaj de programare logică adecvat(Arity, Prolog II, Winprolog sau Visual Prolog), atunci nu ne rămâne decât să scriem programul de mai sus şi apoi să-l chestionăm asupra formulelor care ne interesează.

În Prolog II un scop sau o cerere procedurală este marcată prin semnul >; în Arity şi în Winprolog (veziPP2) întrebările sunt prefixate de semnele ?-.

Aşa, de exemplu, pentru formulele scrise la început, în Prolog II vom scrie:

>sau(si(p,q),si(r,s)) ; > si(p, sau(q, şi(r, non(s)))). În limbajul Ppp2 întrebările sunt marcate prin ?- sau(si(p,q),si(r,s)) şi ?- si(p, sau(q, şi(r, non(s)))).

Un program Prolog cu un număr mai mic de instrucţiuni şi mult mai eficace este următorul:

P1 val(0,0).
P2 val(1,1).
P3 val(si(p,q),b):- val(p,u), cont_si(u,q,b).
P4 val(sau(p,q),b):- val(p,u), cont_sau(u,q,b).

P5 val(non(p),b):-val(p,u), bascul (u,b).
P6 cont_si(0,q,0).
P7 cont_si(1,q,b):- val(q,b).
P8 cont_sau(1,q,1).
P9 cont(0,q,b) :- val(q,b).
P10 bascul(0,1).
P11 bascul(1,0).

P1 și P2, aceleași cu R1 și R2, sunt reguli de oprire a programului. Recursivitatea este asigurată, de asemenea, de instrucțiunile P3, P4, și P5. Dar, spre deosebire de R3, R4 și R5, în cazul instrucțiunilor de mai sus sunt evitate unele calcule inutile, cum ar fi, de exemplu, calcularea valorii celui de al doilea argument al unei conjuncții, daca primul are valoarea 0 (vezi instrucțiunile P3 și P6) sau calcularea valorii celui de al doilea argument al unei disjuncții, dacă valoarea primului său argument este 1 (vezi P4 și P8).

A „șterge" un scop sau a rezolva o subproblemă echivalează în Prolog cu a demonstra formula ce-l descrie. Această idee poate conduce la scrierea unui program și mai eficace.

Q1 val(p,1) :-p/.
Q2 val(p,0):-.
Q3 si(p,q):-p,q.
Q4 sau(p,q):-p
Q5 sau(p,q):-q
Q6 non(p):-p/ impas
Q7 non(p):-
Q8 unu:-.

Dacă valoarea lui p este 1, atunci p este adevărat și se stopează atribuirea valorii 0 lui p. Q3-Q5 descriu reguli de inferență întemeiate pe conjuncție (Q3) și pe disjuncție (Q4 și Q5), conform definițiilor acestora: conjuncția este adevărată, dacă ambele ei argumente sunt adevărate; disjuncția poate fi adevărată pe temeiul că este adevărat primul ei argument sau poate fi adevărată pe temeiul că este adevărat cel de al doilea argument al ei. Dacă non p este adevărat, atunci admiterea lui p duce la impas. Ultima instrucțiune vizează tautologii sau formule identic adevărate, care se șterg întotdeauna.

Asupra legăturilor logicii cu programarea logică vom mai reveni în capitolele consacrate logicii predicatelor și teoriei definițiilor și clasificărilor precum și în cel rezervat teoriei argumentării. Vom face, de asemenea, uz de unele noțiuni de programare logică în partea terminală ce cuprinde unele exerciții și rezolvări de probleme.

SECȚIUNEA A II-A

LOGICA PREDICATELOR ȘI BAZELE DE CUNOȘTINȚE

Cap. 1. SINTAXA ŞI SEMANTICA LOGICII PREDICATELOR

1. Întrebări şi probleme

O *problemă* este o asociere între un set structurat de cunoştinţe sau o bază de cunoştinţe transpusă într–un anumit limbaj şi un set de întrebări, astfel alcătuită încât punerea în lucru a cunoştinţelor iniţiale cu ajutorul unor scheme de inferenţă, legi, reguli sau algoritmi să permită aflarea răspunsului la întrebările formulate sau soluţia problemei.

În structura unei probleme intră, aşadar, un *nucleu structurat de cunoştinţe*, o osatură teoretică sau un cadru (frame) conceptual organizat, redată de regulă în limbajul unei discipline ştiinţifice, un sistem de reguli sau scheme logice (cel mai adesea neexprimate explicit şi deţinute latent de rezolvitor sau adresantul problemei) şi un număr de propoziţii interogative, formulate în limbajul bazei de cunoştinţe iniţiale.

Baza de cunoştinţe este alcătuită, la rândul ei, din clauze generice care alcătuiesc cadrul propriu zis al problemei şi dintr–un număr de clauze instanţiate sau factuale care particularizează anumite relaţii sau structuri.

Mai departe, clauzele sunt reductibile la atomi predicativi care descriu proprietăţi sau relaţii. Atomii predicativi, la rândul lor, pot conţine simboluri funcţionale ce descriu operaţii aritmetice.

Esenţial în rezolvarea unei probleme este găsirea unor asignări sau valorizări pentru clauzele generice concordante cu clauzele instanţiate sau de bază şi declanşarea maşinăriei logice sau a procedeelor de calcul sintactice care să conducă la descoperirea soluţiei. Polya spunea că sunt probleme de demonstrat şi probleme de aflat. În cele de demonstrat accentul cade exclusiv pe utilizarea schemelor de inferenţă sau pe aplicarea unor algoritmi, în timp ce în cele de aflat trebuie să utilizăm, în afara unor scheme de inferenţă şi al altor proceduri sintactice, tehnici şi metode de interpretare, asignare şi valorizare a simbolurilor predicative şi semnelor funcţionale care apar în enunţul problemei.

Înţelegerea unei probleme presupune capacitatea rezolvitorului de a–i identifica structura conceptuală, predicatele primare şi conexiunile dintre ele în alcătuirea nucleului conceptual sau a „carcasei teoretice" a problemei, precum şi aptitudinea de a găsi pentru fiecare dintre acestea domenii adecvate de interpretare sau modele, mulţimi constând din obiecte, proprietăţi şi relaţii. Aceasta alcătuieşte, evident, dimensiunea semantică a decodificării enunţului unei probleme.

În mod analog, trebuie înţelese şi raportate la acest cadru *cerinţele* sau întrebările problemei. Întrebările în cazul problemelor de aflat sunt, de regulă, formule predicative deschise pentru care trebuie găsite asignări de valori acordate variabilelor individuale, astfel încât predicatul în cauză să devină o propoziţie adevărată.

Dar o problemă nu poate fi niciodată rezolvată dacă nu punem în lucru o maşinărie inferenţială, un set de reguli de inferenţă, un procedeu de decizie sau o tehnică rezolutivă. Orice rezolvitor are propriul său stil de interpretare a nucleului teoretic şi propria sa tehnică rezolutivă exersată în activităţile sale intelectuale anterioare.

Formalizarea unei probleme presupune definirea unei opţiuni a rezolvitorului cu privire la modul de simbolizare a datelor problemei, fie ele generice sau factuale.

Formalizarea presupune tranziţia de la limba naturală sau limbajul specializat de disciplină la limbajul logico-sintactic în care va fi prelucrată informaţia despre structura şi datele sau restricţiile specifice ale problemei. Formalizarea este pasul decisiv spre sintaxă şi calcul. Aceasta permite desubiectivizarea, standardizarea şi adesea mecanizarea demersului inferenţial–calculatoriu.

Formalizarea este întotdeauna doar un *moment* într–o strategie cognitivă, nu un scop în sine.

La capătul drumului sintactic are loc o revenire la interpretare, la model sau aplicaţie.

Pentru una şi aceiaşi problemă există, cel mai adesea, modalităţi de analiză, reprezentare şi rezolvare alternative. Rezolvitorul trebuie să–şi apropie datele problemei, să le traducă în limbajul său propriu, să–i asocieze modul său intim de reprezentare, o schemă, un graf, un mod de recodificare şi, desigur, un sistem operaţional, o sintaxă adecvată în cadrul căreia să prelucreze informaţia.

În mod curent, rezolvarea unei probleme e precedată de o etapă analitico–reductivă. Reducem problema iniţială la subprobleme şi acestea la alte subprobleme, până când cădem peste nişte scheme, algoritmi sau tehnici de calcul, cvasiunanim acceptate. A rezolva o

problemă înseamnă a reduce necunoscutul la cunoscut. Din punct de vedere sintactico-semantic a rezolva o problemă înseamnă a extinde baza de date explicită sau modelul parţial iniţial astfel încât să obţinem evaluarea atomilor predicativi marcaţi de semnul întrebării. Regulile de inferenţă şi algoritmii de decizie sunt principalele instrumente sintactice prin care se extinde modelul explicit iniţial (întotdeauna un model parţial) la un model explicit complet, respectiv la un model care cuprinde toţi atomii predicativi din baza de cunoştinţe şi, îndeosebi, cei marcaţi de semnul întrebării, care vor apare la sfârşit ca atomi instanţiaţi şi valorizaţi.

Este o înclinaţie firească a rezolvitorilor de probleme să-şi reprezinte demersul rezolutiv ca o căutare a unui drum între două stări epistemice extreme: starea iniţială, când aceasta deţine datele explicite ale problemei sau baza iniţială de cunoştinţe şi starea finală, când aceasta deţine soluţiile la întrebările sau cerinţele problemei. Între acestea se află mulţimea stărilor intermediare („spaţiul de stări"). Tranziţiile de la o stare la alta sunt efectuate cu ajutorul regulilor de derivare sau „producţiilor" A rezolva o problemă înseamnă, aşadar, a găsi un drum eficace într-un graf care duce, de la o stare *iniţială* conformă cu datele problemei, la o stare *terminală* conformă cu cerinţele sau întrebările problemei.

Din această perspectivă, rezolvarea unei probleme se reduce la o tehnică de căutare într-un spaţiu de stări, în condiţiile satisfacerii unui set de restricţii. În literatura de inteligenţă artificială sunt cercetate diferite strategii de parcurgere a spaţiului de stări: strategia parcurgerii în adâncime, strategia parcurgerii în lăţime, strategii teleologice, semantice, liniare, rezoluţii orientate etc. Important de reţinut este ca nodurile şi arcele unui graf pot fi gândite din perspectiva programatorului ca liste şi operaţii efectuate asupra unor liste şi că, deci, ele descriu treceri, de la formule iniţiale şi formule standard, la stări privilegiate ce descriu răspunsuri la întrebări. Rezolvitorul descrie principalele noduri sau stări intermediare, precum şi regulile au algoritmi prin care pot fi executate aceste tranziţii relevante.

A rezolva o problemă înseamnă, din punct de vedere semantic, a extinde sau dezvolta un model al cadrului conceptual şi al restricţiilor problemei, astfel încât acesta să conţină răspunsuri consistente la întrebările problemei. Pe această cale întrebările ca predicate deschise devin predicate închise sau propoziţii, i.e. soluţii.

În rezolvarea unei probleme, maşinăria inferenţială, calculul şi operaţiile sintactice funcţionează sub control semantic, se particularizează

şi adaptează la restricţiile impuse de clauzele factuale sau asignările de valori atribuite variabilelor individuale care apar în semnele funcţionale sau în simbolurile predicative.

E de prisos să mai adăugăm că rezolvarea unei probleme este întotdeauna corelată cu particularităţile lingvistico–calculatorii ale subiectului rezolvitor. Acesta utilizează doar regulile şi deprinderile de inferenţă şi calcul asimilate în travaliul său intelectual anterior. Sunt şi momente rare, când un rezolvitor descoperă şi foloseşte reguli noi, inedite, create de el pe baza reprezentărilor şi intuiţiilor proprii. Este firesc să admitem stiluri şi modalităţi intelectuale de gândire proprii fiecărui rezolvitor.

Aceste consideraţii metodologice generale despre conceptul de problemă vor fi concretizate în cele ce urmează, după ce vom introduce limbajul logicii predicatelor şi semantica acestuia şi, mai ales, pe parcursul rezolvării problemelor propuse cititorului nostru. Unele consideraţii teoretice despre demersul rezolvării unei probleme logice am făcut şi în prima parte a lucrării de faţă, îndeosebi în paragrafele despre formele normale şi bazele de cunoştinţe şi în cel consacrat metodei interpretărilor semantice succesive. Altele vor fi făcute în partea a IV-a consacrată aplicaţiilor, îndeosebi în paragraful despre corelaţiile dintre întrebări, probleme şi soluţii. Cititorul va găsi, de asemenea, numeroase exemple de formalizare şi rezolvare a unor probleme cu mijloace logice, inclusiv prin alcătuirea unor programe în Prolog.

2. Limbajul logicii predicatelor

Logica predicatelor este o creaţie a ultimului secol.

Creatorii ei sunt, în principal, G.Frege, C.S. Pierce, A.N. Whitehead, B.Russell şi David Hilbert. Limbajul logicii predicatelor s–a născut din nevoia întemeierii logice a demonstraţiilor matematice. Un predicat este o schemă sau o formulă deschisă de forma „ x are proprietatea P", de exemplu, „x este înalt" sau „x se află în relaţia R cu y", de exemplu, „x este coleg cu y", care poate deveni propoziţie adevărată sau falsă, după cum variabilele individuale sunt înlocuite printr–un set de constante sau altele sau după modul în care sunt cuantificate. Admiţând că Petru este înalt şi Ion este scund, atunci substituţia lui „x" prin „Petru" va genera o propoziţie adevărată iar substituţia lui „x" prin Ion va genera o propoziţie falsă.

„x este înalt" este un predicat monadic sau de un singur argument. Vom spune că este de aritate 1. Predicatul „x iubeşte pe y"

este un predicat diadic sau de aritate 2 ce descrie o relaţie definită pe produsul cartezian Oameni × Oameni sau pe produsul cartezian Bărbaţi × Femei şi invers.

Conceptul de predicat din logica predicatelor este diferit de conceptul de predicat din silogistica aristotelică.

Pentru a defini riguros limbajul logicii predicatelor trebuie să definim mai întâi alfabetul acestui limbaj şi apoi regulile sale de bine formare.

Alfabetul logicii predicatelor cuprinde următoarele mulţimi de simboluri sau subalfabete:

$A_c = \{ a, b, c, a_1, b_1 \}$	*constante individuale*
$A_{vi} = \{ x, y, z, x_1, x_2 \}$	*variabile individuale*
$A_F = \{ f, g, h, f_1, ... \}$	*simboluri funcţionale*
$A_P = \{ P, Q, R, P_1, ... \}$	*simboluri predicative*
$A_I = \{ = \}$	*semnul identităţii*
$A_Q = \{ \forall, \exists \}$	*cuantificatori*
$A_{CL} = \{ -, \&, v, \supset, \equiv \}$	*conective logice*
$A_G = \{ (,), [,] \}$	*semne de grupare*

Numim *alfabet* al logicii predicatelor mulţimea de simboluri elementare $A = A_{ci} \cup A_{vi} \cup A_F \cup A_P \cup A_I \cup A_Q \cup A_{CL} \cup A_G$

Vom numi *expresie* o secvenţă de simboluri din alfabetul $A = A_{ci} \cup A_{vi} \cup A_F \cup A_P \cup A_I \cup A_Q \cup A_{CL} \cup A_G$

În mulţimea expresiilor, după modul lor de alcătuire, în conformitate cu anumite reguli de bine formare, distingem: termeni, formule atomare, formule moleculare sau alte formule bine formate şi expresii ce nu sunt formule bine formate.

Tipologia expresiilor poate fi redată prin figura nr.1.

termeni	*formule atomare*	*formule moleculare*	*expresii expresii ce nu sunt bine formate*

Fig. 1. Expresii în logica predicatelor de ordinul I

Propunem acum definiţii constructive pentru conceptele de termen, formulă atomară şi formulă bine formată.

2.1. **Definiţia termenilor**

T1. Dacă α aparţine lui A_{ci} atunci α este un *termen.*

T2. Dacă α aparţine lui A_{vi}, atunci α este un *termen.*

T3. Dacă f aparţine lui A_F şi aritatea lui f, notată prin $\delta(f)$, este n şi $t_1,...,t_n$ sunt termeni, atunci f $(t_1,...,t_n)$ este un *termen.*

Potrivit lui T1 simbolurile a, b, c, a_1,... sunt termeni. Ele ţin loc numelor proprii sau descripţiilor individuale. Aşa, de exemplu, putem avea a = Petru, b = Ion c = 5 etc. Potrivit lui T2, x, y, z, x_1,... vor fi, la fel, termeni, dar termeni cu referinţă ambiguă, neprecizată. Ei ţin loc particulelor din limba naturală, „cineva", „ceva", „cineva", „undeva".

Regula T3 permite construirea termenilor compuşi. Dacă f şi g sunt simboluri funcţionale monadice ce ţin loc unor operaţii unare, cum ar fi, de exemplu, y este tatăl lui x, y = f (x) sau z este primul născut al lui x, z = g (x), atunci f (x) şi g (x) vor fi, de asemenea, termeni.

Dacă f_1 desemnează adunarea şi g_1 înmulţirea ca operaţii binare în **N** (se înţelege $\delta(f) = \delta(g_1) = 2$), atunci g (f (x,y), z) va desemna tot un număr din **N** rezultat din împărţirea sumei dintre x şi y cu numărul z.

Vom numi *termeni* expresiile scrise în subalfabetele A_{ci}, A_{vi}, A_F în conformitate cu regulile T1, T2, şi T3.

Termenii sunt, aşadar, mijloace de referire sau denumire a unor obiecte. Ei ţin locul numelor proprii şi descripţiilor individuale din limbile naturale.

Strict vorbind, termenii nu sunt formule logic-predicative, ci material de construcţie pentru alcătuirea formulelor.

În formulele atomare termenii vor servi ca argumente pentru simbolurile predicative.

Admiţând alfabetele A_{ci}, A_{vi}, A_F, atunci, conform cu T1 vor fi termeni a, b, c, a_1 şi b_1 şi conform cu T2 vor fi termeni x, y, z, x_1, x_2 şi dacă $f \in A_F$ şi $\delta(f) = 1$ şi $\delta(g) = 2$, atunci vor fi termeni compuşi: f(a), g(f(a), b), f(g(f(a), b)), g(f(g(f(a), b)), c) etc.

Se observă că un termen compus poate intra în continuare în structura altui termen compus.

Constantele individuale, de exemplu, a, b, c, a_1 pot fi privite drept simboluri funcţionale degenerate sau de aritate 0. Când un semn funcţional f, are aritatea zero i.e. $\delta(f) = 0$, atunci f decade într-o constantă individuală. Un termen compus poate fi privit ca o formă funcţională. Aşa cum în algebră definim unele numere cu ajutorul unor operaţii algebrice, tot aşa în logica predicatelor definim unii termeni compuşi cu ajutorul simbolurilor funcţionale. Un termen

compus este o aplicare a unui simbol funcţional de aritate n la alţi n termeni anterior definiţi. Procedeul este iterativ.

Este util să menţionăm că în scrierea termenilor din logica predicatelor putem utiliza trei moduri de scriere prefixată, infixată şi postfixată.

În scrierea prefixată se scrie la începutul expresiei simbolul funcţional. În scrierea infixată simbolul funcţional se scrie la mijloc, ca + în aritmetică, 3 + 2; dimpotrivă, în scrierea postfixată se scriu întâi argumentele şi la urmă, în dreapta, simbolul funcţional.

2.2. **Definiţia atomilor.**. Atomii sau formulele atomare sunt formule predicative elementare. Un atom este o aplicare a simbolului predicativ sau constantei predicative la un număr corespunzător de termeni.

P1. Dacă P este un simbol predicativ de aritate n, i.e. $\delta(P) = n$, şi $t_1,...,t_m$ sunt termeni atunci $P(t_1,...,t_m)$ este un *atom*.

P2. Dacă t_1 şi t_2 sunt termeni, atunci $t_1 = t_2$ este un *atom* sau o *formulă atomară.*

Numim *atomi* sau *formule atomare* expresiile formate în conformitate cu regulile P1 şi P2.

Exemple

Q(a), P(x), R(y, z), S(f(a), g(y)) sunt formule atomare.

La fel, vor fi atomi predicativi expresiile: y = 2x + 4, a = f(b).

Paradoxal, ceea ce numim atom este din punct de vedere sintactic un agregat, un complex. Sub raport semantic atomii descriu proprietăţi sau relaţii elementare cărora li se pot atribui valori de adevăr. Sunt cele mai simple formule din logica predicatelor despre care putem spune că sunt adevărate sau false. Termenii nu au valoare de adevăr, ci doar semnificaţie sau denotat. În structura atomilor nu apar niciodată conective logice, ci doar simboluri predicative, constante şi variabile individuale şi uneori semne funcţionale.

2. 3. Formule

P3. Atomii sunt formule bine formate în logica predicatelor.

P4. Dacă α este o formulă bine formată în logica predicatelor, atunci $\forall x\alpha(x)$ sau $\exists x\alpha(x)$ vor fi, de asemenea, formule bine formate.

P5. Dacă α şi β sunt formule bine formate în logica predicatelor, „-" este semnul negaţiei şi * este un conectiv logic binar

din A_{CL}, adică $\wedge, \vee, \supset$ sau $\equiv$, atunci $-\alpha$ şi $\alpha * \beta$ vor fi formule bine formate în logica predicatelor.

Numim *formule bine formate* în logica predicatelor formulele formate în conformitate cu regulile P1–P5. Limbajul logicii predicatelor este mulţimea tuturor formulelor bine formate, respectiv mulţimea tuturor formulelor ce pot fi obţinute cu ajutorul alfabetului A, dat mai sus, prin aplicarea, de un număr finit de ori a regulilor P1–P5

Exemple

Drept exemple de formule bine formate putem considera expresiile:

P(a), R(a, b), R(x, b), Q(f(a), z), S(x, y, z), R2(x, g(x, z), h(a)) $\supset$ (P(x) & Q(g(x, z)) & T(h(a))), $\forall$zQ(f (a), z), $\exists x \forall y \forall z$S (x, y, z)

Numim *propoziţie* o formulă logic-predicativă în care apar numai constante individuale sau în care toate variabilele sunt legate. Astfel în enumerarea de mai sus sunt propoziţii P(a), R(a, b), $\forall$zQ(f (a), z), $\exists x \forall y \forall z$S(x, y, z). Sunt formule predicative deschise sau *predicate* toate celelalte. In formula $\forall$zQ(f (a), z), z în $\forall$z este variabilă sub cuantificator, iar z în corpul formulei se numeşte *variabilă legată* de cuantificatorul universal $\forall$z. Expresia R2(x, g(x, z), h(a)) $\supset$ (P(x) & Q(g(x, z)) & T(h(a))) este un predicat de două variabile x, z.

Un *predicat* este o formulă deschisă care are cel puţin o variabilă liberă.

Simbolurile care apar într-o formulă pot fi *simboluri logice* şi *simboluri nelogice*. In rândul *simbolurilor logice* intră variabilele individuale, *x, y, z,* conectivele logice (-, $\wedge$, v, $\supset$, $\equiv$), cuantificatorii ($\forall$, $\exists$) , simbolul egalităţii (=), parantezele sau semnele de grupare şi virgula (,). Tot simboluri logice vor fi, desigur, valorile de adevăr, 1 şi 0, şi desigur operatorii modali L , M care apar în logicile modale etc. *Simboluri nelogice* vor fi constantele individuale, simbolurile funcţionale, variabilele predicative şi, în particular, variabilele propoziţionale care pot fi interpretate ca variabile predicative de aritate 0. Simbolurile +, -, *, / sunt simboluri funcţionale şi, prin urmare, simboluri nelogice. Simbolul egalităţii poate fi văzut, deopotrivă, ca un simbol logic, dar şi ca un simbol predicativ de două argumente. În metalogică un limbaj va fi definit ca o mulţime numărabilă de simboluri nelogice, iar o propoziţie dintr-un limbaj va fi o formulă bine formată în care toate simbolurile nelogice(constantele individuale, simbolurile funcţionale, variabilele predicative) vor fi simboluri din acel limbaj.

Un limbaj important pentru metateoria logicii este limbajul aritmeticii

$L = \{ 0, ', +, * \}$, pe care îl vom prezenta în capitolele de metalogică.

Un termen t este *liber* pentru variabila x în formula α, dacă nici o apariţie liberă a variabilei x în α nu apare în domeniul de acţiune al unui cuantificator $\forall y$ sau $\exists y$, unde y este o variabilă ce apare în termenul t. Astfel termenul y este liber în raport cu variabila x din formula P(x), dar nu este liber în raport cu variabila x din formula $\forall yP(x)$, căci, prin substituirea lui x cu y în $\forall yP(x)$, un argument anterior liber ar deveni, după substituţie, legat.

Prin substituirea unei variabile individuale dintr–o expresie printr–un termen ce conţine şi el variabile, trebuie ca nici o variabilă din acesta să nu devină legată în formula obţinută prin substituţie, într–un context în care ea era liberă în formula iniţială.

Observaţii

Obs. 1. Este important de reţinut că, în limbajul logicii predicatelor, atomii predicativi sunt, cumva, analogi variabilelor propoziţionale; ei pot fi legaţi prin conective logice, tot aşa cum, în logica propoziţiilor, variabilele propoziţionale pot fi legate prin conective logice, generând astfel structuri propoziţionale complexe. Prin legarea atomilor predicativi prin conective logice obţinem structuri predicative complexe sau propoziţii complexe.

Obs. 2. Atomii predicativi sunt instrumente formale ce permit descrierea proprietăţilor şi relaţiilor.

Obs. 3. Atomii predicativi pot fi instanţiaţi sau neinstanţiaţi. Un atom instanţiat nu conţine variabile libere; el are ca argumente pentru simbolul predicativ doar constante individuale şi va descrie, sub raport semantic, o propoziţie, ce poate fi adevărată sau falsă. Un atom neinstanţiat nu este o propoziţie, ci doar o structură aptă de a deveni propoziţie, printr-o interpretare adecvată sau prin cuantificare. Trecerea de la atomii neinstanţiaţi la atomi instanţiaţi corespunde trecerii de la abstract la concret sau determinării şi aceasta se face prin aplicarea regulilor substituţiei în logica predicatelor de ordinul întâi.

Obs. 4. Atomii predicativi sunt „cărămizile" sau componentele elementare care intervin în procesul formalizării logice a cunoştinţelor dintr-un anumit domeniu de cunoaştere sau de activitate practică.

3. Semantica logicii predicatelor

Pentru a defini o semantică a logicii predicatelor trebuie să asociem constantelor individuale, simbolurilor predicative, simbolurilor funcţionale şi variabilelor individuale din alfabetul logicii predicatelor obiecte sau structuri de obiecte dintr–un domeniu de interpretare oarecare. Aceasta o vom face, desigur, cu ajutorul unor funcţii *de interpretare* care proiectează elementele subalfabetelor menţionate mai sus în domeniul de interpretare. Fie D domeniul de interpretare, Ic o funcţie care asociază fiecărui simbol funcţional din A_F de aritate n o funcţie Ic(f), definită pe produsul D^n cu valori în D şi care asociază, de asemenea, fiecărui simbol predicativ din A_P de aritate m o funcţie Ic(P), definită pe D^m cu valori în mulţimea valorilor de adevăr $\mathbf{B} = \{0, 1\}$. Fie Iv o funcţie care asociază oricărei variabile individuale din A_{vi}, un element din D.

Atunci, o *interpretare* este o tripletă v = (D, Ic, Iv), unde D, Ic şi Iv sunt definite ca mai sus. O interpretare corect definită asociază valori de adevăr oricărei formule F din limbajul logicii predicatelor. Convenim să notăm valoarea de adevăr asociată unei formule F prin v(F). Regulile de interpretare vor asocia, de asemenea, termenilor din limbajul logicii predicatelor obiecte individuale din D. Dacă t este un termen, atunci v(t) va fi un obiect din D, numit adesea *denotat* sau *referent* al lui t. Denotatul este, aşadar, obiectul denumit de un termen într-o interpretare semantică dată.

Dăm acum reguli de interpretare pentru constantele individuale, variabilele individuale, simbolurile funcţionale şi pentru simbolurile predicative.(vezi A Thayse Ed, *From Standard Logic to Logic Programming*, John Willey & Sons 1988).

I1. Dacă a este o constantă individuală, atunci v(a) va fi un obiect individual din D (Acest caz coincide situaţiei când $\delta(f) = 0$, i.e. aritatea simbolului funcţional f este 0 şi în limbaj apar constante individuale ca a, b, a_i etc).

I2. Dacă x este o variabilă liberă, atunci $v(x) =_{def} Iv(x)$.

I3. Dacă f este un simbol funcţional de aritate n şi $t_1,...,t_n$ sunt termeni, atunci

$v(f(t_1,...,t_n)) = Ic(f)(v(t_1),..., v(t_n))$.

I4. Dacă P este un simbol predicativ de aritate m şi $t_1,...,t_m$ sunt termeni, atunci: $v(P(t_1,...,t_m)) =_{df} Ic(P)(v(t_1),..., v(t_m))$.

I5. Dacă t_1 şi t_2 sunt termeni, atunci $v(t_1 = t_2) = 1$ (sau V de la verum, adevărat în latină), dacă şi numai dacă, $v(t_1) = v(t_2)$; în mod analog, $v(t_1 = t_2) = 0$(sau fals, F), dacă şi numai dacă: $v(t_1) \neq v(t_2)$.

I6. Dacă α şi β sunt formule bine formate în logica predicatelor, atunci:

$\alpha \wedge \beta$, $\alpha \vee \beta$, $\alpha \supset \beta$, $\alpha \equiv \beta$ vor fi, de asemenea, formule bine formate în logica predicatelor de ordinul întâi, iar valoarea lor de adevăr va fi descrisă de definiţiile conectivelor logice în cauză. În consecinţă, semantica formulelor moleculare sau compuse din logica predicatelor se va defini sub forma unei inducţii matematice, pornind de la valoarea de adevăr a propoziţiilor atomare şi ţinând cont de definiţiile conectivelor logice ce intră în alcătuirea lor.

Aceste reguli sunt suficiente pentru interpretarea formulelor din logica predicatelor care nu conţin cuantificatori sau a „matricelor" din logica predicatelor.

Dacă M ($x_1,...,x_n$) este o „matrice" având variabile $x_1,...,x_n$, atunci v(M) va desemna funcţia ce asociază oricărui n–tuple de forma $< v(t_1),..., v(t) >$, valoarea de adevăr v(M ($t_1,..., t_n$)).

Trebuie acum să dăm reguli de interpretare pentru formulele ce conţin cuantificatori. Pentru aceasta, introducem, mai întâi, o convenţie de notare. Dacă $v = (D_I, I_c , I_v)$ este o interpretare, x este o variabilă individuală, şi d este un element din D_I, atunci $v_{x/d}$ va desemna interpretarea v_1, astfel că $D_{v1} = D_v$, $Iv1(c) = Iv(c)$, $Iv_1(x) = d$ şi $Iv1(y) = Iv(y)$, pentru orice variabilă y diferită de x.

Cu alte cuvinte, interpretarea v_1 va fi identică cu interpretarea v, şi, deci, va avea acelaşi domeniu de interpretare şi aceleaşi funcţii de interpretare pentru simbolurile funcţionale şi pentru toate simbolurile predicative, cu singura deosebire că în v_1 i se atribuie lui x semnificaţia d, unde $d \in D_I$, interpretarea tuturor celorlalte variabile individuale rămânând aceeaşi ca în v. Interpretarea v_1 este o extindere conservativă a interpretării v. Ea păstrează toate asignările interpretării anterioare şi, în plus, dă valori unei noi variabile, celei aflate sub cuantificator.

Dăm acum regulile de interpretare pentru cuantificatori.

I7. Dacă α este o formulă şi x este o variabilă, atunci $v(\forall x\alpha(x)) = 1$, dacă $v_{x/d}(\alpha)=1$ pentru *orice* element d din D.

I8. Dacă α este o formulă şi x este o variabilă, atunci $v(\exists x\alpha(x)) =1$, dacă $v_{x/d}(\alpha) = 1$ pentru cel puţin *un* element d din D.

O formulă α din logica predicatelor este *adevărată* într-o interpretare v, dacă $v(\alpha) = 1$. Se mai spune că formula satisface interpretarea v sau că v este un model al lui α. Aceasta se scrie uneori astfel:

$$v \models \alpha$$

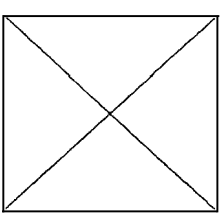
O formulă α este *validă*, într–o interpretare v sau v–*validă*, dacă şi numai dacă este adevărată în orice atribuire de valori Iv, dată variabilelor individuale din interpretarea v = (D, Ic, Iv).

O formulă α este *realizabilă* într–o interpretare v sau v–*realizabilă*, dacă şi numai dacă există cel puţin o atribuire de valori dată variabilelor individuale din Iv în care α devine adevărată.

O formulă α este *infirmabilă,* dacă are cel puţin un *contra-model* respectiv, dacă există cel puţin o atribuire de valori în Iv pentru care α devine falsă.

O formulă α este *irealizabilă* sau *contradicţie logică* într–o interpretare v sau v–*irealizabilă,* dacă nu există nici o atribuire de valori dată variabilelor individuale care să o satisfacă, i.e. în care să devină adevărată.

Raporturile dintre cele patru concepte definite mai sus pot fi reprezentate intuitiv cu ajutorul relaţiei abstracte de pătrat logic definită de noi în paragraful 11 din prima secţiune.



Între cele patru concepte semantice au loc relaţiile bine cunoscute din logica aristotelică şi generalizate de noi în paragraful menţionat, relaţiile de contrarietate (incompatibilitate), subcontrarietate (disjuncţie-slabă), contradicţie (disjuncţie tare sau exclusivă) şi subalternare (implicaţie)

Definirea semantică a conceptului de formulă adevărată permite, între altele, definirea relaţiei de consecinţă logică.

Spunem că propoziţia B este o *consecinţă logică semantică* din propoziţiile A1,...,An, dacă şi numai dacă, în orice interpretare v şi în orice atribuire de valori φ dată variabilelor individuale în care sunt adevărate propoziţiile A1,...,An va fi adevărată şi propoziţia B. Relaţia de consecinţă logică semantică se notează prin $\models$.

$$A1, A2,..., An \models_{\varphi} B \Leftrightarrow \forall v \forall \varphi (v \models_{\varphi} A1 \& ... \& An \supset v \models_{\varphi} B)$$

Definirea conceptului de model sau (interpretare în care o formulă este adevărată), ca şi definirea noţiunilor de realizabilitate, validitate, infirmabilitate şi irealizabilitate sau inconsistenţă va conduce la o mai corectă înţelegere a tipologiei problemelor.

In plus, în raport cu limbajele de programare şi cu strategiile de inteligenţă artificială, distincţiile de mai sus, pe lângă înţelesul lor epistemologic indiscutabil, dobândesc un interes metodologic şi operaţional fiind intim legate de definirea cerinţelor sau scopurilor în limbajul de programare Prolog.

4. Predicate şi propoziţii. Întrebări şi răspunsuri

Am afirmat mai sus că, din punct de vedere sintactic, un *predicat* este o formulă deschisă, mai exact o formulă care conţine cel puţin o variabilă liberă. Din punct de vedere semantic, un predicat este o formulă interpretabilă într–un domeniu de obiecte prin înlocuirea variabilelor cu obiecte din acel domeniu (mai exact cu numele lor) şi prin determinarea faptului dacă dintre diferitele n–tupluri de obiecte din acel domeniu există unele care satisfac relaţia asociată simbolului predicativ în cauză. Un *predicat* este, din punct de vedere semantic o formulă *aptă de a deveni adevărată sau falsă;* o formulă valorizabilă.

Predicatul nu este o propoziţie, ci o schemă sau o structură aptă de a deveni o propoziţie. Un predicat nu trebuie confundat cu simbolul (sau simbolurile) predicative prin care este exprimat.

Fiind dat un predicat, dispunem de două modalităţi de a crea sau obţine propoziţii: prin *cuantificare* şi prin *instanţiere* sau asignare de valori variabilelor individuale.

Fie $P(x_1,...,x_n)$ un predicat. Atunci din acesta putem obţine $2^n \times n!$ propoziţii determinate prin cuantificarea variabilelor $x_1,...,x_n$ Fie C(x, y, z) predicatul „x comunică lui y, z" Prin cuantificare putem obţine din acest predicat $2^3.3! = 8.\ 1.\ 2.\ 3 = 48$ de propoziţii în care variabilele x, y, şi z sunt cuantificate complet.

Este util să reamintim că, atunci când cuantificăm, spre deosebire de scrierea curentă, care are loc de la stânga la dreapta, scrierea cuantificatorilor se face de la dreapta spre stânga. Să presupunem că formula are n variabile libere. Pentru introducerea primului cuantificator vom avea $2 \times n$ alternative. Mai departe, pentru introducerea celui de al doilea cuantificator, vom avea $2 \times n–1$ alternative. Pentru introducerea ultimului cuantificator, vom avea $2 \times n–(n–1)$, respectiv 2×1 alternative Numărul total de alternative va fi, în cele din urmă, $2^n.n!$ sau $2^n \times n!$

Prin cuantificarea unei singure variabile într-un predicat acesta devine un predicat de n–1 variabile. Astfel, predicatul deschis de mai sus C(x,y,z), prin cuantificarea existenţială a variabilei individuale z, $\exists z C(x,y,z)$, devine predicatul diadic $C^2(x,y)$ = „ x comunică lui y" Dacă, dimpotrivă, vom scrie $\exists x C(x,y,z)$, obţinem predicatul diadic C^2 (y, z) = „ lui y i s–a comunicat z". Dacă cuantificăm existenţial pe y, $\exists y C(x,y,z)$ obţinem predicatul diadic C^2 (x, z) = „x comunică z" Dacă asupra acestor predicate diadice continuăm cuantificarea vom obţine predicate monadice sau proprietăţi. Astfel $\exists y \exists z C(x,y,z)$ este un predicat monadic $C^1(x)$ = „x comunică" In sfârşit, dacă scriem $\exists x \exists y \exists z C(x,y,z)$, obţinem propoziţia „Cineva a comunicat cuiva ceva", care evident, într–un anumit context, poate fi adevărată sau falsă.

Tocmai pentru că predicatele, ca formule deschise, nu reprezintă angajări asertorice, ele fac obiectul operatorilor interogativi. Într–un predicat deschis *întrebarea* poate viza unul sau altul dintre argumentele funcţiei logice predicative. Astfel în predicatul C(x,y,z)=„x comunică lui y, z" noi putem fi interesaţi să cunoaştem emitentul, adresantul (sau receptorul) sau conţinutul comunicării (mesajul). Pentru fiecare dintre aceste cazuri putem să formulăm o întrebare adecvată.

(?x) C (x, y, z)	Cine a comunicat lui y, z ?
(?y) C (x, y, z)	Cui i–a comunicat x, z ?
(?z) C (x, y, z)	Ce i–a comunicat x lui y ?

Răspunsul la întrebare va extrage din datele explicite ale bazei de cunoştinţe constanta individuală şi respectiv obiectul corespunzător din domeniul de interpretare.

Dacă, de exemplu, suntem interesaţi într–un anume subiect al comunicării, într–o anume problemă concretă c, i.e. z = c, atunci putem, mai departe, să ne întrebăm cine a mai comunicat cu cineva în privinţa subiectului c.

(?x) (?y) C (x, y, c)

În ipoteza existenţei unei baze de cunoştinţe adecvate calculatorul ne va furniza o listă de persoane astfel încât prima i–a comunicat celei de a doua ceva despre problema c Să presupunem că am implementat într–un program Prolog o bază de cunoştinţe descriind predicatele „bărbat(x), femeie(y) părinte(x,y), căsătorit(x,y). Clauzele factuale vor avea forma „Ion este bărbat", „Ana este femeie" „Ion este părintele lui

Dan", „Ana este un părinte al lui Dan", „Ion este căsătorit cu Ana", „Dan este căsătorit cu Maria", „Dan este părintele lui Barbu", „Dan este părintele Elenei", Barbu este părintele lui Petru".

Intr–o astfel de circumstanţă putem fi interesaţi să aflăm dacă „Este sau nu Ion părintele lui Barbu ?"

? –părinte (Ion, Barbu).

? –părinte (Ana, Elena).

(De reţinut că în Prolog numele proprii sunt constante individuale şi se scriu cu literă mică iar variabilele individuale se scriu cu literă mare !).

Acestea sunt *întrebări de tip da – nu,* când ni se cere de fapt să ne pronunţăm asupra valorilor de adevăr ale unui atom instanţiat, dată fiind o bază de cunoştinţe factuale şi anumite clauze generice exprimând relaţii dintre clase de obiecte. Pentru răspunsul la astfel de întrebări se vor utiliza demonstratoare de teoreme întemeiate pe derivări sau respingeri rezolutive.

Putem, evident, să ne întrebăm, dacă Ion este sau nu părintele lui Barbu, sau ce fel de relaţii de rubedenie sunt între ei. În acest caz se pune problema definirii unor concepte derivate, date fiind un număr limitat de concepte admise ca primare. Problema este, în cazul considerat, cum vom defini relaţiile derivate de copil, mamă, tată, fiu, fiică, unchi, mătuşă, strămoş, descendent, rudă, rude în linie dreaptă, rude colaterale, strămoşi comuni ai unor indivizi, văr, noră, ginere, socru mare, soacră mică, cumnat, cuscru, etc., utilizând pentru aceasta exclusiv relaţiile primare admise. Apoi, o dată definite aceste relaţii, cum identificăm indivizii care satisfac respectivele relaţii de rubedenie?

În cazul introducerii unor noi concepte, avem de a face cu cerinţa de a construi definitori într–un limbaj dat, pentru concepte intuitiv clare într–un alt limbaj. Între definiendum şi definiens sau definitor trebuie să se asigure relaţia de echivalenţă referenţială (vezi în volumul 2, *Teoria definiţiilor*). În plus, procedurile sintactice utilizate trebuie să conserve această relaţie de echivalenţă şi să fie finite şi controlabile. Un loc central în acest scop îl ocupă definirea adecvată a unor relaţii de recurenţă, cum este, de exemplu, pentru cazul considerat relaţia ancestral sau strămoş.

Dacă relaţiile în cauză au fost corect definite, atunci baza de cunoştinţe poate fi chestionată cu folos asupra noţiunilor nou introduse, punând o lungă listă de întrebări de forma:

?–bunic (ion, X).	%	Cui îi este Ion bunic %
? –fiică (elena, ion)	%	Este Elena fiica lui Ion %
? –strămoş (Z, petru), bărbat (Z)	%	Care sunt strămoşii de sex bărbătesc ai lui Petru ? %
? –soacră (Y, ana)	%	Cine este soacra Anei? %

Răspunsurile la astfel de probleme extrase din baza de cunoştinţe vor avea forma:

X = barbu
pentru prima întrebare din cele patru de mai sus,
no
pentru a doua,
Z = barbu;
Z = dan;
Z = ion;
pentru cea de a treia întrebare, şi
Y = maria,
pentru cea de a patra întrebare.

O *soluţie* la o întrebare sau scop epistemic într–un program Prolog este o asignare de valori dată variabilelor unui predicat deschis ce descrie întrebarea ce–l transformă pe acesta într–o propoziţie derivabilă din baza de cunoştinţe dată (A se vedea în acest sens şi teoria ecuaţiilor logice prezentată anterior).

Sunt întrebări care pot avea mai multe soluţii sau răspunsuri adevărate şi întrebări care nu au nici o soluţie, respectiv nici un model sau interpretare adecvată.

Fără a intra în detalii tehnice, este util, deocamdată, să reţinem că putem întâlni întrebări de tip da/nu, în care ni se cere să ne pronunţăm asupra valorii de adevăr a unei propoziţii date. În acest caz ne întrebăm: este formula F demonstrabilă ca teoremă din baza de cunoştinţe dată S ? Dacă din S derivă rezolutiv F, atunci răspunsul nostru este pozitiv. Dacă, dimpotrivă, S&–F este realizabilă, i.e S&–F. are un model, atunci răspunsul nostru este nu.

La întrebările de forma (?x)P(x) răspunsul este pozitiv, dacă există, în domeniul de interpretare D, cel puţin un individ d, desemnat

prin numele propriu a, astfel încât P(a) este adevărat şi negativ dacă nu există un astfel de individ. Evident, pentru (?x,y)R(x,y) răspunsurile vor fi alcătuite din perechi de elemente ce satisfac R.

Problema definibilităţii unui predicat derivat P, date fiind conceptele primare B1,...Bn, poate fi descrisă într–o primă instanţă ca:

$$(?P)\ (P \equiv f_B(B_1,\ldots,B_n)$$

unde ≡ desemnează, aici, relaţia de echireferenţialitate sau identitatea de semnificaţie, f_B este o funcţie logic propoziţională şi $B_1,\ldots,B_n$ sunt atomi predicativi de bază. (A se consulta, în acest sens, partea a III-a din lucrarea noastră *Logica simbolică şi bazele de cunoştinţe*, vol 2, Universitatea „Politehnica" Bucureşti 1999, consacrată teoriei definiţiilor şi clasificărilor).

Întrebarea devine un scop epistemic sau o cerinţă în Prolog. Răspunsul la o întrebare este întotdeauna dat printr–o procedură.

Clauzele generice din baza de cunoştinţe slujesc ca reguli de prelucrare sau producţiuni.

Nu întâmplător oricărei set de clauze Horn putem să-i asociem un program în Prolog. Vom ilustra aceasta în cazul rezolvării unor probleme concrete.

CAP. 2. METODE DE DECIZIE ÎN LOGICA PREDICATELOR

A *decide* asupra unei formule înseamnă a determina printr–un procedeu oarecare dacă formula este realizabilă sau irealizabilă, validă sau invalidă.

În mod curent oamenii raţionează în cadrul unei limbi naturale sau în cadrul limbajului specializat al unei discipline ştiinţifice. Dar limbile naturale nu sunt lipsite de ambiguităţi. De aceea, pentru a putea verifica validitatea unui raţionament formulat într-o limbă naturală sau într-un limbaj specializat trebuie, mai întâi, să-l formalizăm într-un limbaj logic, de regulă în limbajul logicii propoziţiilor sau în limbajul logicii predicatelor, şi după aceea să decidem asupra formulelor obţinute printr–o metodă adecvată.

Aici ne propunem să prezentăm câteva metode de decizie eficiente în logica predicatelor.

După cum se ştie (A. Church, 1936), limbajul logicii predicatelor este semidecidabil: pentru orice raţionament valid formalizat în limbajul logicii predicatelor există un procedeu formal finit prin care putem atesta validitatea raţionamentului. Pot, în schimb, exista formule invalide pentru care nu putem determina, într-un număr finit de paşi, invaliditatea lor.

1. Metoda tablourilor semantice

Este o metodă apagogică sau prin reducere la absurd. Se emite ipoteza că formula dată spre testare este infirmabilă, respectiv că aceasta va avea un contramodel. Dacă această ipoteză duce la consecinţe contradictorii, atunci formula supusă testării nu are contramodel sau nu este infirmabilă şi, deci, va fi validă. Dacă, dimpotrivă, ipoteza că formula iniţială este infirmabilă nu duce la contradicţii, atunci procedeul ne va furniza o interpretare în care formula iniţială devine infirmabilă, respectiv un contramodel al acesteia. Contramodelul formulei ce descrie raţionamentul va fi, totodată, proba irefutabilă a faptului că raţionamentul din limba naturală de la care am plecat este invalid.

Analiza formulei se face în conformitate cu opt reguli de inferenţă pentru conectivele logic–propoziţionale $\wedge$, $\vee$, $\supset$, $\equiv$ şi patru reguli de inferenţă pentru introducerea şi respectiv excluderea celor doi cuantificatori.

Am prezentat în partea I-a (vezi capitolul 2.4, pp79-80) regulile de analiză logico-semantică a conectivelor logice: $\wedge$, $\vee$, $\supset$ şi $\equiv$. Adăugăm la acestea regulile cuantificatorilor, specifice logicii predicatelor:

Regulile negaţiei cuantificatorilor

$-\forall$ 1	0
	1. $\forall xA$
2.$\exists x-A$	

$-\exists$ 1	0
	1. $\exists xA$
2.$\forall x-A$	

Regulile excluderii cuantificatorilor

E$\exists$ 1	0
1...a...	
2...b...	
3.$\exists xA$	
4. A(c)	

E$\forall$	1	0
	1...a...	
	2...b...	
	3. $\forall xA$	
	5. A(a)	
	6. A(b)	

Observaţii

1. Regula negării cuantificatorului universal ($- \forall$). Dacă este fals că toţi x au proprietatea A, atunci este adevărat că unii x nu au proprietatea A. Din

$-\forall xA \models \exists x-A.$

2. Regula excluderii cuantificatorului existenţial ($-\exists$). Dacă este fals că există un x care are proprietatea A, atunci este adevărat că nici un x nu are proprietatea A, respectiv toţi x sunt non A .

$-\exists xA \models \forall x\ -A$

3.Regula eliminării cuantificatorului existenţial (E $\exists$). Dacă este adevărat că există un x care are proprietatea A şi mai înainte în alte formule au apărut constantele individuale a şi b, atunci există un obiect individual c, diferit de a şi diferit de b, astfel încât c are

proprietatea A. Aceasta este regula instanţierii sau a identificării unui obiect cu o anumită proprietate într–un univers de discurs dat.

4. Regula eliminării cuantificatorului universal (E∀). Dacă este adevărată formula ∀xA şi a şi b desemnează obiecte din domeniul de interpretare D despre care a fost vorba în formulele anterioare, atunci vor fi adevărate în mod necesar A(a) şi A(b). Mai mult, regula eliminării cuantificatorului universal rămâne încă activă, şi în cazul apariţiei unei noi constante individuale (de exemplu, prin utilizarea regulii eliminării cuantificatorului existenţial), aceasta se va aplica şi la constanta individuală nou apărută.

La excluderea cuantificatorului existenţial instanţierea are loc pentru *anumiţi* indivizi din domeniu ce satisfac proprietatea sau relaţia desemnată prin simbolul predicativ, în timp ce la excluderea cuantificatorului universal instanţierea are loc pentru *toţi* indivizii din domeniul D de referinţă.

Aplicând regulile de mai sus unui raţionament, acesta este *valid*, dacă ipoteza infirmabilităţii sale duce în toate alternativele la contradicţii şi *invalid*, dacă ipoteza infirmabilităţii sale se confirmă, respectiv analiza se încheie prin găsirea unui contramodel sau a unei interpretări în care toate premisele sunt adevărate şi concluzia este falsă.

Exemplu

Să considerăm raţionamentul:

1. Toţi inginerii sunt practicieni.
2. Toţi matematicienii sunt teoreticieni.
3. Nici un teoretician nu e practician.

---

4. Nici un matematician nu e inginer.

Formalizăm raţionamentul în limbajul logicii predicatelor, admiţând codificările:

I(x) = x este inginer;
P(x) = x este practician;
M(y) = y este matematician;
T(y) = y este teoretician.

Obţinem:

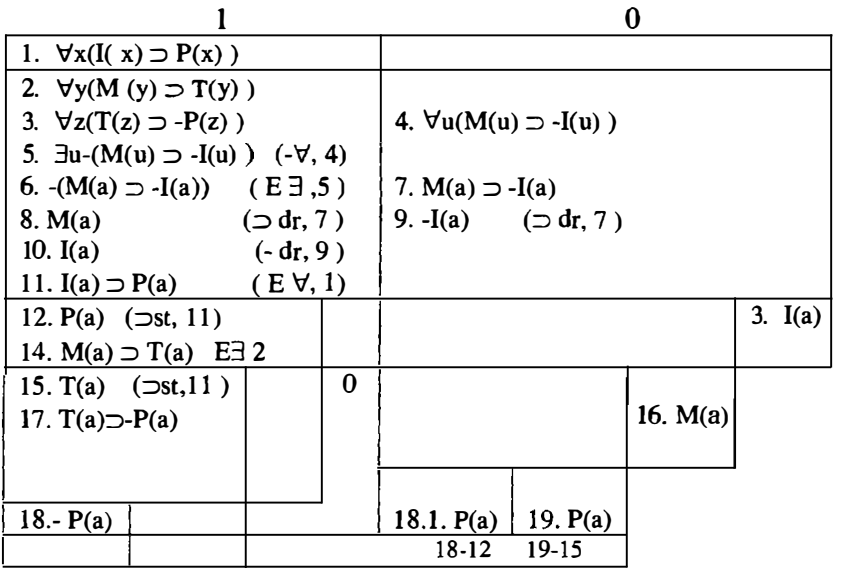
1. $\forall x(I(x) \supset P(x))$
2. $\forall y(M(y) \supset T(y))$
3. $\forall z(T(z) \supset \neg P(z))$

---

4. $\forall u(M(u) \supset \neg I(u))$

Metoda tablourilor semantice ne cere să admitem ca adevărate premisele 1, 2, 3 şi să contestăm concluzia 4, respectiv să admitem negaţia acesteia.

Dacă aplicând cele 12 reguli de analiză (8 dintre acestea au fost prezentate în prima secţiune, paragraful 2.4), ajungem în orice subtablou la contradicţii sau la *închiderea* subtabloului, atunci raţionamentul este valid. Dimpotrivă, dacă cel puţin un tablou nu se închide atunci raţionamentul are contraexemplu şi nu este valid.


1
0
1. ∀x(I(x) ⊃ P(x))
2. ∀y(M (y) ⊃ T(y))
3. ∀z(T(z) ⊃ -P(z))
5. ∃u-(M(u) ⊃ -I(u)) (-∀, 4)
6. -(M(a) ⊃ -I(a)) (E ∃ ,5)
8. M(a) (⊃ dr, 7)
10. I(a) (- dr, 9)
11. I(a) ⊃ P(a) (E ∀, 1)
4. ∀u(M(u) ⊃ -I(u))
7. M(a) ⊃ -I(a)
9. -I(a) (⊃ dr, 7)
12. P(a) (⊃st, 11)
14. M(a) ⊃ T(a) E∃ 2
3. I(a)
15. T(a) (⊃st,11)
17. T(a)⊃-P(a)
0
16. M(a)
18.- P(a)
18.1. P(a)
19. P(a)
18-12
19-15

Tablou semantic în logica predicatelor

Câteva explicaţii vor fi, probabil, utile.

Formulele 1, 2, 3 scrise în domeniul adevărului i.e. în stânga, respectiv în domeniul adevărului şi 4 în domeniul falsului i.e. în 0, exprimă ipoteza infirmabilităţii formulelor ce descriu raţionamentul. Formula 5 provine din 4 prin regula falsităţii unei formule cuantificate universal sau regula negaţiei cuantificatorului universal. Formula 6 provine din 5 prin regula excluderii cuantificatorului existenţial. Formula 7 provine din 6 prin legile negaţiei. Mai departe, 8 şi 9 provin din 7, prin regula implicaţiei în domeniul falsului. 10 provine din 9

prin legile negaţiei. 11 derivă din 1 prin regula excluderii cuantificatorului universal, în domeniul adevărului.

Prima scindare a tabloului este provocată de analiza formulei 11, care este o implicaţie adevărată. Adevărul ei se poate datora faptului că are consecventul adevărat (vezi 12) sau faptului că are antecedentul fals (vezi 13). Formula 14, ca şi 11, provine din ipotezele iniţiale, în speţă din 2, prin aplicarea regulii excluderii cuantificatorului universal. Analiza lui 14 generează scindarea tabloului iniţial în tablourile I1, cu 15 adevărat şi în I2, cu 16 fals.

Formula 17 provine din 3, tot prin regula excluderii cuantificatorului universal, iar analiza mai departe a lui 17 generează scindarea tabloului I1 în tabloul I1a şi I1b în care se scriu –P(a) în domeniul adevărului şi T(a) în domeniul falsului.

Un tablou se închide, dacă una şi aceiaşi formulă apare, atât în domeniul adevărului, cît şi în domeniul falsului.

Este util să precizăm că un subtablou „moşteneşte" formulele tabloului din care a provenit şi că, deci, se poate închide un tablou oarecare, de exemplu, I2, ce conţine formula M(a) în domeniul falsului (vezi 16) cu simetricul său I2, din domeniul adevărului, ce nu conţine direct nici o formulă, dar are într–un tablou anterior, în tabloul originar, aceiaşi formulă M(a), în domeniul adevărului(vezi 8). Închiderea unui tablou a fost marcată cu linie dublă orizontală, atât în domeniul adevărului, cît şi în domeniul falsului, sub care au fost scrise numerele perechilor opuse de atomi.

E de prisos să mai adăugăm că analiza unui set de formule iniţiale provenite din formalizarea unui raţionament (mai exact din ipotezele acestuia şi negaţia concluziei) se încheie când am ajuns peste tot la formule atomare.

2. Metoda arborilor de decizie

Este o metodă de testare a realizabilităţii formulelor în logica propoziţiilor şi în logica predicatelor. A verifica realizabilitatea unei formule înseamnă a căuta sistematic un model pentru acea formulă. Dacă reuşim să descoperim un model, atunci formula este realizabilă.

a. Regulile metodei arborilor de decizie în logica predicatelor

Vor fi, în continuare, valabile α - regulile şi β - regulile introduse anterior pentru logica propoziţiilor (vezi Partea I-a , cap.4.1, pp 107). La acestea se vor adăuga încă două clase de reguli: reguli pentru eliminarea

cuantificatorului universal (sau γ-reguli) şi reguli pentru eliminarea cuantificatorului existenţial sau δ - reguli.

În afara acestora se vor utiliza încă două reguli pentru eliminarea negaţiei din faţa unui cuantificator.

Antecedent	Consecvent
$-\forall x\ P(x)$	$\exists x\ -P(x)$
$-\exists x\ P(x)$	$\forall x\ -P(x)$

Regulile eliminării negaţiei unui cuantificator

Formula din antecedent este echivalentă cu cea din consecvent. Dacă nu-i adevărat că există cineva care să aibă proprietatea P, atunci este adevărat că toţi (i.e. nici unul dintre ei) nu au (nu are) acea proprietate şi invers, dacă este fals că toţi indivizii dintr-un anumit domeniu au o anumită proprietate, atunci este adevărat că unii nu o au.

Scriem, în continuare, regulile eliminării cuantificatorului universal şi pe cele ale eliminării cuantificatorului existenţial.

Excluderea cuantificatorului existenţial are prioritate asupra excluderii cuantificatorului universal.

Antecedent	Consecvent
γ	γ(a)
$\forall x\ A(x)$	A(a)
$\forall x\ -A(x)$	-A(a)

γ- reguli sau eliminarea cuantificatorului universal

Antecedent	Consecvent
δ	δ(a)
$\exists x\ P(x)$	P(a)
$\exists x\ -P(x)$	-P(a)

δ- reguli sau eliminarea cuantificatorului existenţial

Regula excluderii cuantificatorului universal spune că ceea ce este valabil despre toţi este valabil despre cazurile anterior instanţiate

şi despre cele care vor mai apare ulterior. La nivelul drumului într-un arbore, regula poate fi schematizată astfel:

1. ..... a .....
2. ..... b .....
3. $\forall x P(x)$

---

4. P(a)
5. P(b)

Dacă în nodurile 1 şi 2 au apărut constantele individuale a, b şi în nodul 3 formula $\forall x P(x)$, atunci putem conchide, mai jos, că P(a) şi P(b) şi .....

Regula excluderii cuantificatorului universal este relativă la un număr de instanţieri şi rămâne deschisă pentru alte instanţieri viitoare. Formula $\forall x P(x)$, într-un fel, rămâne şi după ce a fost instanţiată prin câteva cazuri.

Dimpotrivă, regula excluderii cuantificatorului existenţial este definitivă, absolută şi ea îngăduie eliminarea formulei cuantificată existenţial.

La nivelul unui drum într-un arbore, regula eliminării cuantificatorului existenţial poate fi redată astfel:

1. ..... a ....
2. b
3. $\exists x P(x)$

4. P(c)

$c \neq a, c \neq b$

Dacă în etichetele nodurilor anterioare au apărut formule având constantele individuale a şi b şi la un moment dat apare formula $\exists x P(x)$ care spune că există un individ x cu proprietatea P, atunci vom denota individul cu proprietatea P printr-o constantă individuală diferită de indivizii menţionaţi în formulele anterioare. De exemplu:

1. elev (ion)
2. student (dan)
3. există X hoţ (X)

4. hoţ (pavel)

pavel ≠ ion, pavel ≠ dan şi pavel trebuie să fie numele unui hoţ.(Am urmat, mai sus, regulile de scriere ale instrucţiunilor Prolog, nu gramatica limbii române).

Nu putem conchide pripit că Ion sau Dan sunt hoţi, doar pe temeiul că aceştia au fost anterior menţionaţi, în altă ordine de idei. Iar despre individul Pavel vom conchide că este hoţ, dacă şi numai dacă instanţiind propoziţia „există un individ X astfel încât hoţ (X)" în contextul dat îl găsim pe individul „Pavel" ca satisfăcând predicatul „hoţ" Aceasta presupune, desigur, o investigaţie penală adecvată.

A susţine că există un x astfel că P(x), $\exists x\, P(x)$ se reduce la a găsi în domeniul de referinţă D un exemplu sau un caz ce satisface P.

b. Construirea arborilor de decizie în logica predicatelor

Sunt presupuse condiţiile de terminare ale arborilor şi regulile α şi β formulelor specificate în capitolul 4.1 pentru logica propoziţiilor. La acestea se adaugă regulile pentru cuantificatorul universal şi pentru cel existenţial:

Regula cuantificatorului universal E∀:

Dacă A este o γ - formulă (ca $\forall x A1(x)$) ce apare în eticheta unui nod f, creează un nod succedent f', astfel ca:

$$E(f')=E(f)\cup\{\gamma(a)\}$$

unde a este o constantă individuală ce apare în f.

Regula cuantificatorului existenţial E∃:

Dacă A este o δ - formulă (ca $\exists xP(x)$) ce apare în eticheta unui nod f, creează un nod nou f ' ca succesor al nodului f, astfel că:

$$E(f')=E(f)-\{A\}\cup\{\delta(a)\}$$

unde a este o constantă individuală ce nu apare în formulele ce etichetează nodul f.

Observaţii:

Obs. 1. Se observă că în trecerea de la un nod f la un nod f ' cu ajutorul regulii eliminării cuantificatorului universal E∀ sau γ - regulilor, *nu se şterge formula cuantificată universal A, ci se adaugă doar la aceasta instanţieri ale predicatului prin constantele individuale ce apar în etichetele nodului f.*

Exemplu de aplicare a regulii E∀:

$$E(f)=\{\forall x\, Q(x), P(a), R(a, b)\}$$

$$E(f')=\{\forall x\, Q(x), Q(a), Q(b), P(a), R(a, b)\}\ E\forall$$

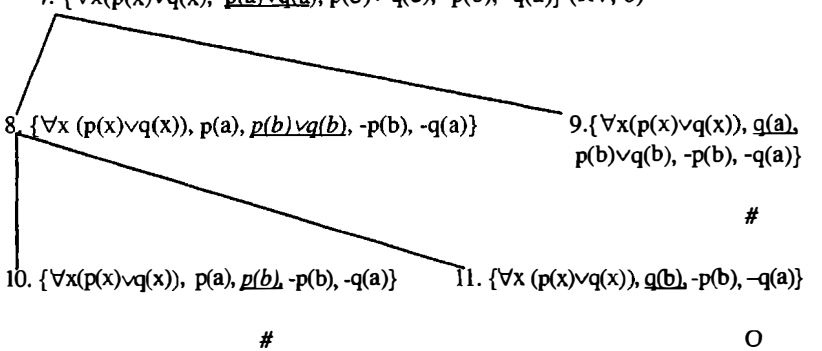
Obs. 2. Dimpotrivă, în aplicarea regulii cuantificatorului existenţial E∃ sau δ - regulii în trecere de la nodul f la nodul succesor f ', respectiv de la E(f) la E(f '), *dispare formula cuantificată existenţial A* şi în locul ei apare o instanţiere a acesteia printr-o constantă individuală diferită de toate constantele individuale anterioare.

Obs. 3. Nici E∀ sau γ- regulile şi nici E∃ sau δ- regulile nu duc la scindarea tablourilor. Acestea se comportă ca α - regulile.

Vom nota, în continuare, simbolurile predicative cu litere mici, ca în programarea logică, în loc de litere majuscule. Vom scrie p(x), r(z,y) în loc de P(x) şi R(z, y).

Exemplul 3. $\forall x(p(x) \vee q(x)) \supset (\forall x p(x) \vee \forall x q(x))$,

1. $-(\forall x(p(x) \vee q(x)) \supset (\forall x\, p(x) \vee \forall x\, q(x)))$
2. $\{\forall x(p(x) \vee q(x)), -(\forall x\, p(x) \vee \forall x\, q(x))\}$
3. $\{\forall x(p(x) \vee q(x)), -\forall x\, p(x), -\forall x q(x)\}$
4. $\{\forall x(p(x) \vee q(x)), \exists x\text{-}p(x), \exists x\text{-}q(x)\}$
5. $\{\forall x(p(x) \vee q(x)), \exists x\, \text{-}p(x), \text{-}q(a)\}$ (E∃, 4)
6. $\{\forall x(p(x) \vee q(x)), \text{-}p(b), \text{-}q(a)\}$ (E∃, 5)
7. $\{\forall x(p(x) \vee q(x), \underline{p(a) \vee q(a)}, p(b) \vee q(b), \text{-}p(b), \text{-}q(a)\}$ (R∀, 6)



Observaţii

Obs. 1. Demn de reţinut este faptul că la punctele 5-6 s-a aplicat regula E∃ sau δ-regula şi că aceasta a impus pentru eliminarea fiecărui cuantificator existenţial introducerea a câte unei constante individuale inedite. În rândul 5, la eliminarea lui $\exists x\, \text{-}p(x)$, s-a ales b distinct de a, introdus pentru eliminarea lui $\exists x\, \text{-}q(x)$.

Obs. 2. Formula din exemplul 3 este infirmabilă; nu este validă. A se observa, în acest sens, eticheta nodului terminal 11, marcată prin cerculeţ şi contraexemplul inserat de acesta. Cititorul poate verifica singur acest fapt. Pentru un domeniu D={a, b}, antecedentul $\forall x(p(x) \vee q(x))$ este adevărat, deopotrivă, pentru x=a şi pentru x=b, în timp ce consecventul $\forall xp(x) \vee \forall xq(x)$ este fals în interpretarea <q(b), -p(b), -q(a)>, căci -q(a) infirmă pe $\forall xq(x)$ şi -p(b) infirmă pe $\forall xp(x)$.

Obs. 3. Dimpotrivă, conversa ei, $\forall xp(x) \vee \forall xq(x) \supset \forall x(p(x) \vee q(x))$ este validă.

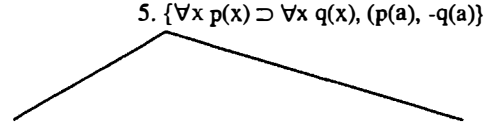
Obs. 4. De reţinut că în 7 am subliniat disjuncţia care provoacă scindarea în nodurile 8 şi 9. La fel am scris subliniat, în 8 şi 9, argumentele disjuncţiei din 7. Mai departe, am scris *cursiv* disjuncţia din 8, care provoacă scindarea în nodurile 10 şi 11. La fel, am scris cursiv argumentele acesteia, în apariţiile lor din 10 şi 11.

Exemplul 4. Să se verifice validitatea formulei:

$F_2 = (\forall x\, p(x) \supset \forall x\, q(x)) \supset \forall x\, (p(x) \supset q(x))$.

Pentru aceasta testăm, prin metoda arborilor de decizie, negaţia lui F_2

1. $-((\forall xp(x) \supset \forall xq(x)) \supset \forall x(p(x) \supset q(x))$
2. $\{\forall xp(x) \supset \forall xq(x)), -\forall x(p(x) \supset q(x))\}$
3 $\{\forall xp(x) \supset \forall xq(x)), \exists x -(p(x) \supset q(x))\}$
4 $\{\forall x\, p(x) \supset \forall x\, q(x)), \exists x\, (p(x) \wedge -q(x))\}$
5. $\{\forall x\, p(x) \supset \forall x\, q(x), (p(a), -q(a)\}$



6. $\{-\forall x\, p(x), (p(a), -q(a)\}$
8. $\{\exists x\, -p(x), (p(a), -q(a)\}$
0. $\{-p(b), (p(a), -q(a)\}$

O

7 $\{\forall xq(x), (p(a), -q(a)\}$
9. $\{q(a), -(p(a), -q(a)\}$

#

Formula nu este validă, deoarece are un contramodel CM={-p(b), p(a), q(a)}. Într-adevăr, pentru un domeniu D={a, b} este fals $\forall xp(x)$, deoarece avem -p(b). Deci este fals $\forall xp(x)$ şi este adevărat $-\forall xp(x)$ şi adevărul acestuia duce la adevărul lui

$-\forall x\, p(x) \vee \forall x\, q(x)$ şi, în consecinţă, la adevărul antecedentului F_2.

Pe de altă parte, din p(a) şi -q(a) rezultă falsitatea lui $p(a) \supset q(a)$. Dacă este fals că $p(a) \supset q(a)$, atunci este fals că $\forall x(p(x) \supset q(x))$.

Cum în aceeaşi interpretare este adevărat antecedentul lui F_2 şi fals consecventul lui F_2 şi F_2 este o implicaţie, rezultă că este falsă implicaţia F_2.

Raţionamentul de mai sus poate fi redat schematic:

a) $-p(b) \Rightarrow -\forall x\, p(x) \Rightarrow -\forall x\, p(x) \vee \forall x\, q(x) \Rightarrow \forall x\, p(x) \supset \forall x\, q(x)$

b) $p(a) \wedge -q(a) \Rightarrow -(p(a) \supset q(a)) \Rightarrow -\forall x(p(x) \supset q(x))$

c) $\forall x p(x) \supset \forall x q(x) \wedge -\forall x(p(x) \supset q(x)) \Rightarrow -((\forall x p(x) \supset \forall x q(x)) \supset \forall x(p(x) \supset q(x)))$

Simbolul $\Rightarrow$ nu este implicația ci o relație de consecință logică.

Exemplul 5. Să se decidă, prin arbori de decizie, asupra formulei

$F_3 = \forall x(A(x) \supset B(x)) \supset (\forall x A(x) \supset \exists x B(x))$.

Pentru aceasta testăm, cu ajutorul regulilor α, β, γ, δ, negația formulei în cauză.

1 $-(\forall x(A(x) \supset B(x)) \supset (\forall x A(x) \supset \exists x B(x))$

2 $\{\forall x(A(x) \supset B(x)), -(\forall x A(x) \supset \exists x B(x))\}$ $(-\supset, 1)$

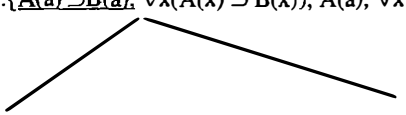
3 $\{\forall x(A(x) \supset B(x)), \forall x A(x), -\exists x B(x)\}$ $(-\supset, 2)$

4 $\{\forall x(A(x) \supset B(x)), \forall x A(x), \forall x\text{-}B(x)\}$ $(\exists, 3)$

5 $\{\forall x(A(x) \supset B(x)), \forall x A(x), -B(a), \forall x\text{-}B(x)\}$ $(E\forall, 4, III)$

6. $\{\forall x(A(x) \supset B(x)), A(a), \forall x A(x), -B(a), \forall x\text{-}B(x)\}$ $(E\forall, 5, II)$

7. $\{\underline{A(a) \supset B(a)}, \forall x(A(x) \supset B(x)), A(a), \forall x\, A(x), -B(a), \forall x\, -B(x)\}$ $(E\forall, 6)$



8 $\{\underline{-A(a)}, \forall x(A(x) \supset B(x)), \underline{A(a)}, \forall x A(x), -B(a), \forall x\, -B(x)\}$

#

9 $\{\underline{B(a)}, \forall x(A(x) \supset B(x)), A(a), \forall x A(x), \underline{-B(a)}, \forall x\text{-}B(x)\}$ $(E\supset, 7)$

#

Formula F_3 este validă.

Tranziția de la 3 la 4 s-a făcut pe baza regulii de eliminare a negației cuantificatorului existențial ($-\exists x A(x) \equiv \forall x\, -A(x)$). Tranzițiile (4 -5), 5-6, (6-7) s-au efectuat pe baza regulii eliminării cuantificatorului universal $E\forall$ sau a regulii instanțierii. Se observă că formula-părinte ce conține cuantificatorul universal nu dispare; ea permite doar producerea unei instanțe și rămâne potențial activă.

Trecerea de la 7 la 8 și 9 este rezultatul aplicării unei β-reguli, și anume a regulii excluderii unei implicații adevărate.

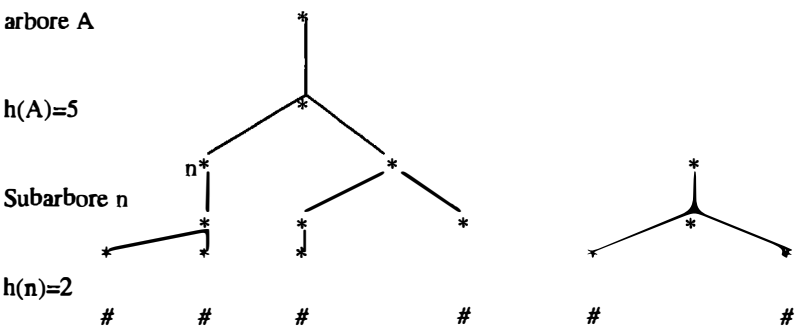
c. Corectitudinea metodei arborilor de decizie în logica predicatelor

Teorema 5. *Teorema consistenței. Enunțul teoremei. Fie F o formulă bine formată în logica predicatelor, exprimând o propoziție și A_F arborele ei de decizie complet. Dacă A_F este un arborele închis, atunci F este o formulă irealizabilă.*

Reformularea teoremei. Enunţul teoremei se reformulează ca un enunţ mai tare: Dacă un subarbore A îşi are rădăcina într-un nod n şi este închis, atunci formulele ce etichetează nodul n sunt irealizabile (unsatisfiabile).

Se înţelege, dacă F≡-G, atunci G va fi o formulă validă. Pe această cale demonstraţia pe care o vom da devine o demonstraţie de noncontradicţie sau consistenţă.

Intuiţia de bază o reprezintă arborele de decizie, ca o construcţie încheiată, şi mai ales maniera în care se construieşte aceasta, cu ajutorul celor patru tipuri de reguli. Regulile de tip α, γ şi δ nu duc la scindări. Singurul tip de reguli care duc la scindări sunt β - regulile. (de tip $\vee$).



Demonstraţia se construieşte prin inducţie pe înălţimea h a nodului n.

Demonstraţie.

Cazul de la bază. Dacă h = 0, atunci n este o frunză. Deoarece A_F se închide, eticheta lui n, E(n) va conţine o pereche opusă de laterali şi, deci, va fi irealizabilă (n este de fapt şi rădăcină şi frunză!).

Dacă h > 0, atunci pentru a se ajunge de la nodul n la nodul fiu n' s-a făcut uz de cel puţin una dintre regulile α, β, γ sau δ.

Considerăm pe rând cele patru cazuri.

Cazul 1. Pentru a se ajunge la n' a fost utilizată o α - regulă ($\wedge$, -$\vee$, -$\supset$, -$\subset$).

Atunci etichetele nodului părinte şi nodului fiu vor fi:

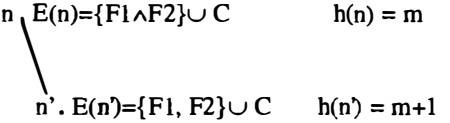
E(n)={F1$\wedge$F2}$\cup$ C "părinte"
E(n')={F1, F2}$\cup$ C "fiu"

unde C este o mulţime, posibil vidă, de formule „pasive"

Din D10 (vezi partea I, cap 4.1) rezultă că înălţimea fiului este cu o unitate mai mare decât a părintelui.

$fiu(n, n') \supset (h(n) = m \supset h(n') = m+1)$

Tranziţia de la n la n' mediată de o α - regulă poate fi reprezentată:



Cum subarborele având rădăcina în n se închide la n', eticheta E(n') va fi o mulţime de formule irealizabilă şi, deci, va exista în E(n') o formulă A', astfel ca v(A') = 0.

Formula falsă se va afla în mod necesar într-una dintre cele trei subformule din E(n'):

a) A'= $C_0 \subset$ C, i.e. este o subformulă din C şi

b) A'= F1 sau v(F1) = 0. Prin definiţia conjuncţiei, dacă este fals un argument al ei, va fi falsă conjuncţia în întregul ei.

$v(F_1) = 0 \supset v(F1 \wedge F2) = 0$

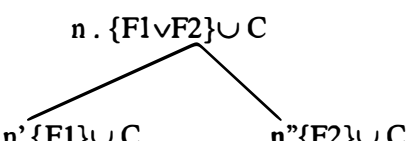
c) A' = F2 sau v(F2) = 0.

Prin aceeaşi argumentare ajungem la falsitatea lui F1∧F2 i.e. v(F1∧F2) = 0.

Prin contrapoziţie, fiecare din cele trei alternative conduce la falsitatea sau irealizabilitatea lui E(n), respectiv a formulei ce etichetează rădăcina subarborelui n.

Cazul 2. Presupunem că a fost utilizată o β regulă pentru realizarea trecerii de la nodul n la nodurile n' sau n".

Grafic ne putem reprezenta aceasta:



Etichetele nodului părinte şi ale nodurilor fii vor fi:

E(n)={F1∨ F2}∪ C
E(n')={F1}∪ C
E(n")= {F2}∪ C.

Cum n' şi n" sunt noduri terminale într-un arbore închis, etichetele lor, E(n') şi E(n"), conţin fiecare câte o pereche opusă de literali şi deci, sunt irealizabile.

Fie v o interpretare arbitrară. Putem avea două alternative:

a) $v(F_0) = 0$ pentru $F_0 \in C \subseteq E(n')$.

Atunci, $v(E(n').) = 0$ şi prin contrapoziţie $v(E(n)) = 0$.

b) Chiar dacă $v(F_0) = 1$, aceasta nu conduce la veridicitatea lui E(n), căci pentru adevărul acestuia ar trebui să fie adevărată cel puţin una dintre etichetele lui n' sau n'', E(n') sau E(n'') Dar acestea sunt, prin ipoteză, irealizabile şi deci trebuie să fie adevărat că:

$v(F1) = v(F2) = 0$.

De unde, prin definiţia lui $\vee$ rezultă pentru subformulele F1, F2 din E(n) că $v(F1 \vee F2) = 0$.

În consecinţă, E(n) va fi irealizabil.

Cazul 3. Pentru efectuarea tranziţiei în cauză, a fost utilizată o γ - regulă, respectiv o instanţiere a unui cuantificator universal. În acest caz vom avea:

$E(n)=C\cup\{\forall x\ F(x)\}$ (părinte)

$E(n')=C\cup\{\forall x\ F(x), F(a)\}$ (fiu)

Prin ipoteza inducţiei E(n') este irealizabilă.

Fie v funcţia de evaluare a unei interpretări arbitrare I. Dacă $v(F(a)) = 0$, atunci $v(\forall x\ F(x))$ este falsă, conform regulii:

Dacă $v(F(a)) = 0$, atunci $v(\forall x\ F(x)) = 0$. (Aceasta este schema inferenţei prin subalternare din logica clasică, de la falsitatea lui I la falsitatea lui A).

Deoarece interpretarea I a fost una arbitrară, eticheta rădăcinii lui n, E(n) va fi irealizabilă.

Cazul 4. Admitem că pentru efectuarea tranziţiei a fost efectuată o regulă de tip δ. Atunci etichetele nodurilor între care se face tranziţia vor fi:

$E(n)= C \cup\{\exists x\ F(x)\}$ (părinte)

$E(n')= C \cup\{F(a)\}$ (fiu)

unde constanta individuală a este inedită şi nu apare în U(n).

Să presupunem, prin absurd, că C ar fi realizabilă. Atunci trebuie să existe o interpretare:

$I=(D, \{R_1, ..., R_n\}, \{d_1, ..., d_k\})$

unde D este un domeniu de obiecte, $\{R_1, ..., R_n\}$ este o listă de literali instanţiaţi, iar $\{d_1, ..., d_k\}$ sunt constante individuale desemnând obiecte din D, astfel că:

$v(\exists x\ F(x)) = 1$ şi deci, pentru un $d \in D$, $v(F(d)) = 1$.

Acum considerăm o extindere a interpretării I la o interpretare:

$I'=(D,\{R_1, ..., R_n\}, \{d_1, ..., d_k, d\})$

pentru E(n'), unde d va fi denotatul constantei a. Deoarece a nu apare în E(n), ea nu va fi printre constantele $a_1, ..., a_k$, deja valorizate de interpretarea I. Aşadar, v'(F(a)) =1. Evident, valorizarea v' fiind o extindere constructivă a lui v, vom avea v'(C) = v(C) =1 şi la fel v'(E(n')) =1.

Dar prin ipoteza inducţiei E(n') este irealizabilă şi deci E(n') = 0.

Demonstraţia de corectitudine sau întemeiere(soundness) arată că, dacă arborele A se închide, atunci F este irealizabilă. Ceea ce înseamnă că dacă vom pleca de la negaţia formulei -F, adică de la o formă G= -F, vom adeveri validitatea lui G.

Demersul demonstrativ prezentat mai sus este evident valabil despre toţi subarborii terminali ai unui arbore complex şi, în consecinţă, prin inducţie despre orice arbore creat cu ajutorul regulilor de tip α, β, γ şi δ. Demonstraţia de mai sus este valabilă şi pentru arborele iniţial A_F al formulei testate F.

În logica predicatelor, spre deosebire de logica propoziţiilor, ordinea aplicării regulilor de analiză sau reducţie devine importantă şi sunt identificate mai multe strategii eficiente de verificare a realizabilităţii unei formule şi, în consecinţă, de verificare a validităţii negaţiei formulei testate.

Ben Ari de la Haifa propune o tehnică de construcţie a arborilor, introducând etichetări prin liste de formule (liste, nu mulţimi!) şi liste de constante, fiecare dintre ele marcate cu indicatori (pointeri), care precizează stadiul sau momentul analizei, permiţând astfel luarea în considerarea a tuturor subformulelor ce trebuie analizate.

Lincolm Wallen de la Oxford, aprofundând o metodă a lui W. Bibel, a elaborat o caracterizare matricială a drumurilor într-un demers demonstrativ.

În încheiere extindem, la limbajul logicii predicatelor, Teorema 4, după care eticheta unui drum deschis are proprietatea subformulei sau alcătuiesc o mulţime Hintikka.

Teorema 6. *Fie F o formulă din logica predicatelor, A_F arborele ei de decizie, r rădăcina acestuia, f un nod terminal sau o frunză dintr-un drum deschis, E(n) eticheta nodului n, h(n) înălţimea nodului n şi* $M = \cup E(n): h(r) \leq h(n) \leq h(f)$ *mulţimea de mulţimi ce etichetează drumul deschis. Atunci M este o mulţime Hintikka.*

Demonstraţie.

Demonstraţia se dă prin inducţie pe mulţimea regulilor prin care sunt generate etichetele drumului deschis într-un arbore de decizie.

Pentru etichetarea nodurilor terminale sau a frunzelor, ca şi a nodurilor provenite din aplicarea α şi β - regulilor, demonstraţia a fost dată în paragraful 14.4 din prma secţiune (vezi şi Teorema 5).

Mai rămân de tratat cazurile γ şi δ - regulilor.

Cazul γ- regulilor. Fie $A=\forall x\ P(x)$ o subformulă în E(n), cu $h(r) \le h(n) \le h(f)$ şi $\{a_1, ..., a_m\}$ lista constantelor individuale ce apar în formulele ce etichetează nodurile precedente lui n. Atunci eticheta unui nod succedent E(n') va conţine instanţierile $P(a_1)$, ..., $P(a_n)$ şi formula A.

Cazul δ- regulilor. Fie $A=\exists x\ P(x)$ o subformulă din E(n) cu $h(r) \le h(n) \le h(f)$ şi $\{a_1, ..., a_m\}$ lista constantelor apărute în formulele ce etichetează nodurile precedente ale lui n. Atunci vom adăuga în eticheta unui nod succedent n' o nouă formă $A'= P(a_{m+1})$, unde a_{m+1} este o constantă inedită şi vom extinde lista constantelor individuale cu constanta nouă a_{m+1}.

Noua listă va fi $\{a_1, ..., a_m, a_{m+1}\}$.

Obs. 1. Aplicarea regulilor α, β, γ, δ în arborii de decizie generează sistematic subformule şi instanţieri şi conduce la identificarea formulelor primare sau inanalizabile, denumite atomi.

Obs. 2. Atomii sunt susceptibili de evaluare prin valori de adevăr 1 sau 0, respectiv „adevărat" sau „fals"

Teorema 7. *Fie E o mulţime Hintikka. Atunci există un model pentru E sau altfel spus E este realizabilă*

Demonstraţie.

A exista un model pentru E presupune a exista o interpretare I pentru E, iar o interpretare I este o pereche $< D, \psi >$, unde D este domeniul de interpretare, iar ψ este funcţia de de interpretare definită pe alfabetul constantelor individuale, semnelor funcţionale şi simbolurilor predictive cu valori în obiectul din D.

Interpretarea I este definită, dacă fiecărei constante individuale a din E i se asociază un obiect determinat din D, fiecărui simbol predictiv din E de aritate n i se asociază o relaţie de aceeaşi aritate definită pe D.

Fie $C=\{a_1, ..., a_m\}$ mulţimea constantelor individuale care apar în mulţimea de formule E.

Definim acum interpretarea I pentru setul E. Fiecărei constante individuale din C îi vom asocia un obiect individual din D.

Mai departe, fiecărui predicat de m argumente din E îi vom asocia o relație m - ară din R, după regula:

$(d_1, ..., d_m) \in R_i$ dacă $p(a_1, ..., a_m) \in E$ şi $\psi(a_i) = d_i$
$(d_1, ..., d_m) \in R_i$ dacă $-p(a_1, ..., a_m) \notin E$ şi $\psi(a_i) = d_i$
$(d_1, ..., d_m) \notin R_i$ dacă $-p(a_1, ..., a_m) \in E$ şi $\psi(a_i) = d_i$

Cele trei condiții stipulate mai sus satisfac exigențele caracterizării atomilor într-o mulțime Hintikka (vezi condiția 1).

Mai rămâne de arătat, printr-o inducție pe structura formulelor A din E, pe baza definițiilor conectorilor logici şi a cuantificatorilor, că orice formulă din E satisface interpretarea I.

O caracteristică majoră a metodei arborilor de decizie în logica predicatelor este eficiența ei desăvârşită în raport cu toate formulele valide.

Teorema 8. *Teorema de completitudine. Dacă F este o formulă validă din logica predicatelor, atunci putem construi un arbore de decizie închis pentru negația lui F.*

Demonstrație

Fie F, prin absurd, o formulă validă din logica predicatelor, al cărei arbore de decizie pentru -F nu se închide. Atunci există un drum deschis în arborele de decizie complet asociat formulei -F.

Potrivit teoremei T6 drumului deschis din arbore îi este asociată o mulțime Hintikka şi, potrivit teoremei T7, o mulțime Hintikka are întotdeauna un model. Deci formula -F are un model. Dar aceasta contrazice ipoteza inițială după care formula F este o formulă validă.

3. Calcul secvențial în logica predicatelor

Reluăm mai jos calculul secvențial fără regula tranzitivității (*cut free sequent calculus*) pentru logica predicatelor de ordinul întâi. Cititorul este rugat să recapituleze cunoştințele introduse în prima secțiune, capitolul 4, pp 107, unde se introduc regulile pentru conectivele logic-propoziționale.

Sunt presupuse ca asimilate cunoştințele despre limbajul logicii predicatelor de ordinul întâi şi semantica acestui limbaj. Prin a şi b

vom nota parametrii oarecare, iar prin c şi d termeni individuali sau simboluri pentru indivizi.

a. Noţiuni preliminare

Calculul secvenţial este o teorie formală a deducţiei şi a argumentării. În centrul preocupărilor sale se află relaţia de consecinţă logică generalizată „Din A_1 şi...şi A_n se deduce B_1 sau...sau B_n", exprimată sub forma unui secvent.

Redăm, mai jos, regulile cuantificatorilor, specifice aplicării calculului secvenţial la logica predicatelor de ordinul întâi:

Cuantificatorul universal

$$\frac{(\forall \mathbf{st})\ \Gamma, A[c/x] \rightarrow \Delta}{\Gamma, \forall \mathbf{x}\mathbf{A} \rightarrow \Delta} \qquad \frac{\Gamma \rightarrow A[a/x], \Delta}{\Gamma \rightarrow \forall \mathbf{x}\mathbf{A}, \Delta}\ (\forall \mathbf{dr})$$

Cuantificatorul existenţial

$$\frac{(\exists \mathbf{st})\ \Gamma, A[a/x] \rightarrow \Delta}{\Gamma, \exists \mathbf{x}\mathbf{A} \rightarrow \Delta} \qquad \frac{\Gamma \rightarrow A[c/x], \Delta}{\Gamma \rightarrow \exists \mathbf{x}\mathbf{A}, \Delta}\ (\exists \mathbf{dr})$$

Observaţii:

Obs. 1. Simbolurile Γ şi Δ nu intervin efectiv în calcul. Ele simbolizează doar celelalte formule din antecedent şi, respectiv, celelalte formule din succedent.

Obs. 2. Formulele trecute în antecedent sunt considerate toate adevărate, iar cele în succedent, toate false. Importantă pentru aplicarea unei reguli este *formula principală* ce apare în antecedentul concluziei (pentru regulile marcate cu st) sau succedentul concluziei (pentru regulile marcate cu dr).

Obs. 3. Fiecare conectiv sau cuantificator logic are o regulă de introducere a sa în antecedent şi o regulă de introducere a sa în succedent

Obs. 4. Citite de sus în jos, regulile descriu căi de agregare, compunere, sinteză sau construirea de formule complexe din cele atomare şi, totodată, de deducere din „axiome" a „teoremelor" Parcurse în sens invers, de jos în sus, regulile descriu căi de analiză

sau descompunere a agregatelor de formule în atomii constituenți și, totodată, de întemeiere sau argumentare a tezei.

Obs. 5. Formula sau formulele scrise deasupra fiecărei linii reprezintă ipotezele sau premisele regulii, iar formula scrisă sub linie reprezintă concluzia.

Obs. 6. Regulile din stânga st introduc un conectiv logic sau un cuantificator în antecedentul unui secvent; cele din dreapta îl introduc și elimină din succedent. Formula ce conține acest conectiv sau cuantificator se numește formula principală și apare în text subliniată.

Obs. 7. Regulile marcate cu asterisc ∀dr și ∃st trebuie să satisfacă condițiile eigenvariabilelor, respectiv simbolurile ce desemnează parametri individuali trebuie să nu apară în concluziile regulilor.

Obs. 8. Trecerea unei formule din antecedent în succedent (sau consecvent) sau invers se face prin schimbarea semnelor formulei (vezi regulile negației -st și -dr).

Obs. 9. Toate regulile au un singur secvent în concluzie. Regulile logicii propoziționale se împart în α-reguli și β-reguli. Vom numi α-reguli sau de tip conjunctiv regulile care nu duc la scindarea arborelui de demonstrare, respectiv regulile care au un singur secvent și în premise. Vom numi β-reguli sau de tip disjunctiv regulile care au în premise câte doi secvenți și, în consecință, duc la căi alternative în arborele de demonstrare. Vor fi β-reguli regulile ∧dr, ∨st, ⊃st. Vom numi γ-reguli regulile ce introduc sau exclud cuantificatorii universali ∀st și ∃dr și δ-reguli regulile ce introduc sau exclud cuantificatori existențiali ∀dr și ∃st.

Obs. 10. Premisele unei reguli sunt alcătuite în exclusivitate din subformule ale concluziei regulii. Trecerea de la concluzie la premise într-o reducție înseamnă în același timp analiză și reducere la elementar.

Obs. 11. Arborele de reducere generat prin aplicarea Bottom Up a regulilor logice la teza de demonstrat conduce la noduri etichetate prin formule ce au proprietatea de subformule.

b. Reducție și deducție în calculul secvențial al predicatelor

Pentru fiecare dintre cuantificatori avem câte o regulă care privește introducerea și excluderea lui în partea stângă a semnului → și câte o regulă care privește introducerea și excluderea lui în partea dreaptă a semnului →.

După cum am remarcat mai sus, introducerea unui operator sau cuantificator are loc în citirea Top Down a schemelor, iar excluderea acestora în citirea acestora Bottom Up.

Spre deosebire de logica propoziţiilor, în logica predicatelor *contează decisiv ordinea în care sunt aplicate schemele de analiză. Nu orice aplicare corectă a schemelor de inferenţă conduce automat la succes în demonstraţie.*

Problemele substituţiei şi ale unificării literalilor devin în acest caz relevante. Trebuie să ţinem seama de universul constantelor individuale şi de structura şi complexitatea formulelor componente.

În plus, în logica predicatelor *nu toate regulile sunt inversabile fără restricţii.*

Ca şi în cazul logicii propoziţiilor, începutul unei demonstraţii prin calcul secvenţial începe prin a plasa teza de demonstrat în partea dreaptă a semnului derivării:

$$\rightarrow F$$

Demonstraţia se obţine prin aplicarea regulilor de reducere la formule iniţiale şi reuşeşte atunci când secvenţii terminali (ce conţin numai literali) sunt axiome.

Reamintim că o *axiomă* este un secvent terminal ce conţine unul şi acelaşi literal în antecedent şi în consecvent.

Să presupunem că ni se cere să demonstrăm în calculul secvenţial formula F:

$F=(\forall x p(x) \vee \forall x q(x)) \supset \forall x(p(x) \vee q(x))$

1. Plasăm formula F ca succedent al semnului $\rightarrow$:

$\rightarrow \underline{(\forall x p(x) \vee \forall x q(x)) \supset \forall x(p(x) \vee q(x))}$ (1)

Singura regulă aplicabilă la (1) este $\supset$dr. Prin aplicarea ei, de jos în sus, obţinem deasupra liniei de inferenţă secventul (2):

$(\forall x p(x) \vee \forall x q(x)) \rightarrow \forall x(p(x) \vee q(x))$ (2)

ceea ce generează schema:

$\underline{(\forall x p(x) \vee \forall x q(x)) \rightarrow \forall x(p(x) \vee q(x))}$ $\supset$dr (3)

$\rightarrow(\forall x p(x) \vee \forall x q(x)) \supset \forall x(p(x) \vee q(x))$

În cazul secventului (2) sunt aplicabile două reguli $\vee$st şi $\forall$dr. Dăm prioritate primei reguli $\vee$st şi obţinem pe baza lui $\vee$st, care este un β-operator, o scindare în arbore provocată de disjuncţia din antecedentul lui 2:

$\underline{4.1.\ \forall x p(x) \rightarrow \forall x(p(x) \vee q(x))}$; 4.2. $\underline{\forall x q(x) \rightarrow \forall x(p(x) \vee q(x))}$ (4)

Mai departe, atât la 4.1 cât şi la 4.2, se poate aplica regula $\forall$st care permite instanţierea formulelor cuantificate universal prin specii sau literali de bază, dar fără să dispară formula cuantificată universal.

Aplicând local şi scriind doar subformulele implicate obţinem pentru 4.1 şi 4.2:

$\underline{p(a), \forall x p(x)}$ $\quad$ $\underline{q(a), \forall x q(x)}$ (4)

$\forall x p(x)$ $\quad$ $\forall x q(x)$

restul formulelor transcriindu-se fără modificare.

Pe această cale, schema (3) va fi supraetajată după cum urmează:

$\underline{p(a), \forall x p(x) \rightarrow \forall x(p(x) \vee q(x))}$; $\underline{q(a), \forall x q(x) \rightarrow \forall x(p(x) \vee q(x))}$ $\forall$st

4.1. $\underline{\forall x p(x) \rightarrow \forall x(p(x) \vee q(x))}$; 4.2 $\underline{\forall x q(x) \rightarrow \forall x(p(x) \vee q(x))}$ $\vee$st

$\underline{(\forall x p(x) \vee \forall x q(x)) \rightarrow \forall x(p(x) \vee q(x))}$ $\supset$dr (5)

$\rightarrow(\forall x p(x) \vee \forall x q(x)) \supset \forall x(p(x) \vee q(x))$

Este lesne de observat că linia 3 (numărând de jos) a fost obţinută prin aplicare regulii $\vee$st, iar linia 4 prin aplicarea de două ori a regulii $\forall$st.

Este util să observăm că aplicarea regulii $\forall$st nu a dus la eliminarea cuantificatorului universal, ci doar la instanţierea acestuia, el rămânând apt de noi instanţieri.

Pentru încheierea analizei noastre mai rămâne să aplicăm de două ori regula $\forall$dr, respectiv regula infirmării unei propoziţii universale.

Această regulă este marcată de restricţia variabilei speciale (eigenvariable) care interzice ca simbolul prin care se instanţiază variabila să apară în concluzia schemei. Cum în $\forall x(p(x) \vee q(x))$ nu apare constanta *b*, putem să operăm substituţia x/b şi să instanţiem $\forall x(p(x) \vee q(x))$ prin $p(b) \vee q(b)$.

Arborele de reducţie complet va fi:

#	#
$\forall$st 11. $\underline{p(a), p(b), \forall x p(x) \rightarrow p(b), q(b)}$	12. $\underline{q(a), q(b), \forall x q(x) \rightarrow p(b), q(b)}$ $\forall$st,10
$\vee$dr 9. $\underline{p(a), \forall x p(x) \rightarrow p(b), q(b)}$	10. $\underline{q(a), \forall x q(x) \rightarrow p(b), q(b)}$ $\vee$st,8
$\forall$dr,5 7. $\underline{p(a), \forall x p(x) \rightarrow p(b) \vee q(b)}$	8. $\underline{q(a), \forall x q(x) \rightarrow p(b) \vee q(b)}$ $\forall$dr, 6
$\forall$st,3 5. $\underline{p(a), \forall x p(x) \rightarrow \forall x(p(x) \vee q(x))}$	6. $\underline{q(a), \forall x q(x) \rightarrow \forall x(p(x) \vee q(x))}$ $\forall$st,4
$\underline{3. \forall x p(x) \rightarrow \forall x(p(x) \vee q(x))}$;	4. $\underline{\forall x q(x) \rightarrow \forall x(p(x) \vee q(x))}$ $\vee$st,2

$\underline{2. (\forall x p(x) \vee \forall x q(x)) \rightarrow \forall x(p(x) \vee q(x))}$ $\supset$dr

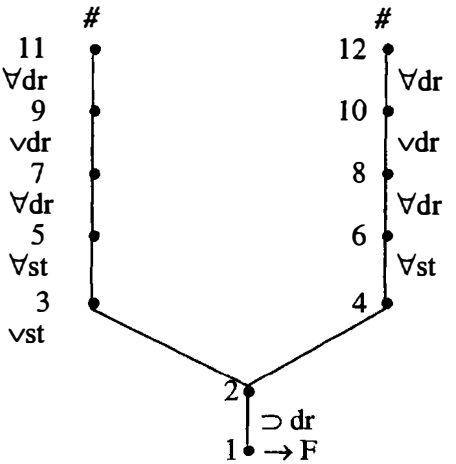
$1. \rightarrow(\forall x p(x) \vee \forall x q(x)) \supset \forall x(p(x) \vee q(x))$

După aplicarea regulii $\forall$dr, pentru încheierea reducerii au mai fost aplicate de câte două ori regulile eliminării disjuncţiei la dreapta, $\vee$dr şi eliminării cuantificatorului universal la stânga, $\forall$st, ţinând cont de instanţierea lui x prin b.

Aplicarea regulii disjuncţiei la dreapta devine intuitivă, dacă considerăm disjuncţia falsă şi din falsitatea ei conchidem asupra falsităţii ambelor sale argumente.

$F(A \vee B) \Rightarrow F(A), F(B)$. La fel: $F(\forall x P) \Rightarrow \exists x F(P(x)) \Rightarrow F(P(a))$, de unde $F(P(a))$ este totuna cu $-P(a)$. ($F(A)$ se citeşte: „Este fals A").

Întregul demers logic analitic reductiv poate fi redat prin arborele de reducţie din figura de mai jos:



Dacă citim schema de reducţie de sus în jos, atunci obţinem o demonstraţie din axiome a tezei de demonstrat.

Este uşor să observăm că secvenţii terminali sunt axiome, respectiv cazuri de aplicare a principiului identităţii i.e. p(b) → p(b), care ne apare generalizat prin aplicarea regulii de slăbire, deopotrivă, la stânga şi, deopotrivă, la dreapta.

Mai departe, trecerea de la 11 la 9 şi de la 12 la 10 se face prin aplicarea regulii de subsumare a clauzelor.

Trecerile de la 9 la 7 şi de la 10 la 8 se fac prin inferenţă de la falsitatea argumentelor unei disjuncţii la falsitatea disjuncţiei ca atare.

Trecerile de la 7 la 5 şi de la 8 la 6 se fac pe baza regulii contraexemplului. Dacă un obiect *b* dintr-un domeniu de referinţe *D* nu are o proprietate *p* sau *q*, atunci va fi falsă propoziţia universală conform căreia *toate* obiectele din acel domeniu au acea proprietate.

Trecerile de la 3 şi 4 la 2 se face pe baza regulii introducerii disjuncţiei în stânga. În sfârşit, trecerea de la 2 la 1 se face prin teorema deducţiei :

Dacă $\Gamma, A \models B$, atunci $\Gamma \models A \supset B$.

Se înţelege că semnul de derivare → se citeşte „se deduce" în lectura Top Down şi „se reduce" în lectura Bottom Up.

Este bine să precizăm că în calculul secvenţial relaţia de consecinţă logică semantică $\models$ este interpretată ca o consecinţă logico-

semantică tare, ca $\models^{v}$ şi nu ca $\models^{t}$, cum este interpretată în metoda arborilor de decizie.(v = valid; t = true)

Cele două relaţii de consecinţă semantică dau rezultate identice ori de câte ori elementele lor constitutive sunt propoziţii. Ele nu mai dau rezultate identice dacă între ipoteze admitem şi formule deschise, cum sunt de exemplu atomii predicativi neinstanţiaţi.

Pentru cititorul ce tânjeşte după exemple concrete, sugerăm să interpreteze formula:

$F=(\forall x p(x) \vee \forall x q(x)) \supset \forall x(p(x) \vee q(x))$ în felul următor:

$p(x)$ = *x este bărbat*, $q(x)$ = *x este femeie*, şi să folosească această interpretare pentru înţelegerea schemelor secvenţiale.

Este uşor de văzut că F se descompune în:

$F_1 = \forall x p(x) \supset \forall x(p(x) \vee q(x))$ şi $F_2 = \forall x q(x) \supset \forall x(p(x) \vee q(x))$, respectiv: *toţi bărbaţii sunt bărbaţi sau femei* şi *toate femeile sunt bărbaţi sau femei.*

La fel se poate lesne observa că inversa lui F nu este validă. Căci este adevărat că *orice* fiinţă umană este bărbat sau femeie, dar fiinţele umane nu sunt *toate* bărbaţi sau t*oate* femei.

Să testăm acum conversa lui F:

$F^{-1} = \forall x(p(x) \vee q(x)) \supset (\forall x p(x) \vee \forall x q(x))$

Postulăm ca în cazul anterior:

$\rightarrow \forall x(p(x) \vee q(x)) \supset (\forall x p(x) \vee \forall x q(x))$ (1)

Aplicând schemele de inferenţă proprii calculului secvenţial la (1) obţinem derivarea:

©

9. $q(a), p(a) \vee q(a), p(b) \vee q(b), \forall x(p(x) \vee q(x)) \rightarrow p(a), q(b)$ $\forall$st,8

\# 8. $q(a), \forall x(p(x) \vee q(x)) \rightarrow p(a), q(b)$ $\forall$st,7

6. $p(a), \forall x(p(x) \vee q(x)) \rightarrow p(a), \forall x q(x)$; 7. $q(a), \forall x(p(x) \vee q(x)) \rightarrow p(a), \forall x q(x)$ $\vee$st,5

5. $p(a) \vee q(a), \forall x(p(x) \vee q(x)) \rightarrow p(a), \forall x q(x)$ $\forall$st,4

4. $\forall x(p(x) \vee q(x)) \rightarrow p(a), \forall x q(x)$ $\forall$dr,3

3 $\forall x(p(x) \vee q(x)) \rightarrow \forall x p(x), \forall x q(x)$ $\vee$dr,2

2. $\forall x(p(x) \vee q(x)) \rightarrow (\forall x p(x) \vee \forall x q(x))$ $\supset$dr,1

1. $\rightarrow \forall x(p(x) \vee q(x)) \supset (\forall x p(x) \vee \forall x q(x))$

Primii patru paşi nu au făcut uz de β-reguli care să ducă la scindare. Prima regulă de acest tip a apărut în al cincilea pas. Ramificaţia din

stânga a calculului se termină printr-o axiomă: *p(a)*, $\forall x(p(x) \vee q(x)) \rightarrow p(a)$, $\forall x q(x)$, marcată prin simbolul #. Dimpotrivă, aripa din dreapta nu se încheie printr-o axiomă, ci printr-o formulă infirmabilă. Într-un univers sau domeniu de interpretare D = {a, b} şi o interpretare I = { -p(a), q(a), p(b), -q(b)}, formula iniţială $F = \forall x(p(x) \vee q(x)) \supset (\forall x p(x) \vee \forall x q(x))$ este infirmabilă.

c. Calculul secvenţial şi arborii de decizie

Calculul secvenţial şi arborii de decizie sunt două metode înrudite.

Ambele sunt întemeiate pe un raţionament apagogic sau prin reducere la absurd. *Presupunem că setul de formule de la care plecăm este infirmabil.* Apoi aplicând un număr de reguli de analiză pentru conectori şi cuantificatori explorăm consecinţele ce pot fi degajate din această ipoteză iniţială. În cazul arborilor, dacă prin aplicarea regulilor ajungem, în toate alternativele, la formule elementare inconsistente, atunci formula sau setul de formule în cauză este neinfirmabil sau valid. În cazul calculului secvenţial, dacă analiza se încheie cu găsirea a cel puţin unui secvent terminal (elementar) ce nu e axiomă, atunci formula are un contramodel şi este, deci, infirmabilă şi raţionamentul pe care îl descrie este invalid.În cazul că analiza *se încheie peste tot cu axiome*, atunci formula testată este *validă.*

Similaritatea persistă şi la nivelul regulilor de analiză. În arborii de decizie avem câte două reguli pentru fiecare conector logic binar. Prima pentru cazurile sale de adevăr; cea de a doua pentru cazurile sale de fals. La fel, în calculul secvenţial avem câte o regulă pentru apariţia conectorului principal în stânga (i.e. pentru adevărul acestuia) şi câte o regulă pentru apariţia acestuia în dreapta (i.e. pentru falsitatea acestora).

Pentru cuantificatori ambele metode dispun de câte patru reguli, îndeplinind roluri similare.

Ambele metode lucrează cu mulţimi de tip Hintikka generate de reguli de tip conjunctiv (α-reguli), de tip disjunctiv (β-reguli), de tip γ şi δ, respectiv pentru cuantificatori universal şi existenţial în domeniul adevărului şi falsului.

Atât calculul secvenţial cât şi metoda arborilor de decizie pot fi utilizate pentru determinarea validităţii sau infirmabilităţii unei formule, respectiv pentru determinarea faptului dacă formula are contramodele. Pentru aceasta, în calculul secvenţial plasăm formula F, pe care vrem să o testăm în dreapta săgeţii: $\rightarrow F$.

Pentru obţinerea aceluiaşi efect, în arborii de decizie pornim de la negaţia formulei F ce descrie raţionamentul, respectiv de la admiterea premiselor şi a negării concluziei, i.e de la -F.

Dar ambele metode pot fi utilizate şi pentru determinarea faptului dacă o formulă sau un set de formule are vreun model, respectiv pentru verificarea realizabilităţii. În calculul secvenţial vom porni, pentru aceasta, de la plasarea formulei F în stânga săgeţii, F→, iar în arborii de decizie vom aplica metodele de analiză direct la formula F a cărei realizabilitate o testăm.

Ambele metode sunt metode semantice de tip Top Down, atunci când le utilizăm în scopuri analitic reductive. Pornim de la evaluarea ipotetică a unei formule complexe şi ajungem la concluzii certe despre existenţa sau inexistenţa unor contramodele ale acesteia.

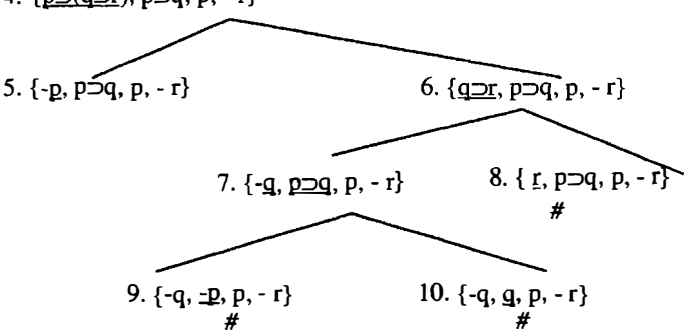
Prin ambele coborâm de la complex la elementar. Ambele pot fi utilizate şi ascendent sau constructiv.

Ambele metode sunt, în principal, de tip analitic-reductiv şi ne permit să identificăm contramodele sau axiome cu ajutorul cărora putem evalua formula analizată sau să identificăm un set de axiome din care aceasta poate fi demonstrată. Ambele pot fi utilizate şi pentru a căuta temeiuri pentru anumite teze sau propoziţii.

Ambele metode generează, ca urmare a aplicării regulilor, *arbori de derivare* care descriu punctele nodale ale demersului analitic reductiv, ca şi demersul sintetic constructiv. Metoda arborilor de decizie este analitică în cazul parcurgerii drumului Top Down, iar calculul secvenţial este analitic în direcţia Bottom Up, de la teza de demonstrat la „axiome" Asemănările dintre cele două metode pot fi observate mai uşor dacă vom considera una şi aceeaşi formulă testată prin cele două metode.

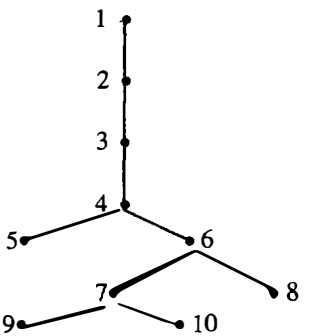
F = (p⊃(q⊃r)) ⊃((p⊃q) ⊃(p⊃r))

1. {-[(p⊃(q⊃r)) ⊃((p⊃q) ⊃(p⊃r))]}
2. {p⊃(q⊃r), - ((p⊃q) ⊃(p⊃r))}
3. {p⊃(q⊃r), p⊃q, - (p⊃r)}
4. {p⊃(q⊃r), p⊃q, p, - r}



	#	#	
	9.p → p, q, r;	10. p, q → q,r #	⊃st,7
#	7. p⊃q,p → q,r	8.p⊃q, p,r → q,r	⊃st,6
5. p⊃q, p → r, p;	6. p⊃q,p, q⊃r → r		⊃st,4
4. p⊃(q⊃r), p⊃q, p → r			⊃dr,3
3. p⊃(q⊃r), p⊃q → p⊃r			⊃dr,2
2 (p⊃(q⊃r)) → ((p⊃q) ⊃(p⊃r))			⊃dr,1
1 →(p⊃(q⊃r)) ⊃((p⊃q) ⊃(p⊃r))			

Redăm mai jos arborele de derivare asociat celor două metode aplicate asupra aceleiaşi formule F:



Demn de reţinut este faptul că unei formule valide nu-i este asociat în mod univoc un arbore de derivare unic, ci o clasă de arbori de derivare având aceiaşi secvenţi terminali închişi sau „axiome" sau aceiaşi secvenţi terminali deschişi sau contraexemple.

Calculul secvenţial este reversibil. Un arbore de decizie poate fi citit Top Down sau demonstrativ, şi Bottom Up sau reductiv, i.e. de la teoremă la axiome.

Secvenţii 5,8,9 şi 10 sunt „axiome"

4. Metoda rezoluţiei

Metoda rezoluţie se aplică unor formule aduse la o formă standard numită formă clauzală. Pentru a înţelege metoda rezoluţiei trebuie să asimilăm mai întâi noţiunile de forme normale prenexe, forme normale Skolem, clauze Davis–Putnam precum şi operaţiile de substituire şi unificare.

a. Clauze

In limbajul comun, o clauză este o propoziţie condiţională de forma:

„dacă A_1, A_2,...A_n, atunci B"

Acesteia îi corespunde în logica propoziţiilor implicaţia:

1. $A_1 \& a_2 \& ... \& A_n \supset B$

care prin schemele:

$\alpha \supset \beta \equiv -\alpha \vee \beta$

$-(\alpha \wedge \beta) \equiv -\alpha \vee -\beta$

d,evine:

2. $-A_1 \vee -A_2 \vee ... \vee -A_n \vee B$

Formula 2 este *clauză Horn pozitivă sau clauză program.*

Orice formulă ce descrie o incompatibilitate este o clauză Horn negativă. De exemplu:

3 $-A \vee -B \vee -C$

este o *clauză Horn negativă.*

Formula 2, care este o clauză Horn pozitivă, dacă n=3, poate fi redată în limbajul de programare Prolog prin instrucţiunea:

4. B :– A1, A2, A3.

şi se citeşte „B este adevărată, dacă sunt adevărate A_1, A_2, A_3" sau „Scopul B este satisfăcut dacă sunt satisfăcute scopurile A_1, A_2, A3"

b. Forme normale conjunctive

Pentru o formulă dată, forma ei clauzală Robinson este tot una cu forma ei normală conjunctivă, FNC. O formă normală conjunctivă este o conjuncţie de disjuncţii elementare. O disjuncţie elementară este o disjuncţie de *atomi* cu sau fără semnul negaţiei înaintea simbolului predicativ. Putem, spune, de asemenea, că o clauză Robinson este o disjuncţie de literali, *literalii* fiind atomi în apariţie pozitivă sau în apariţie negată.

In logica predicatelor aducerea unei formule la formă clauzală presupune transpunerea ei mai întâi în formă normală prenexă, adică într–o formă în care toţi cuantificatorii, dacă există, apar la începutul formulei, în aşa–numitul *prefix*. La rândul ei, forma normală prenexă se aduce la o formă normală Skolem. Aceasta este o formă normală prenexă din care au fost eliminaţi cuantificatorii existenţiali în conformitate cu două reguli de skolemizare.

c. Forme normale prenexe

Pentru a aduce o formulă la forma normală prenexă procedăm astfel:

1. Eliminăm conectivele nebooleene, $\supset$ şi $\equiv$ utilizând echivalenţele:

5. $A \supset B \equiv -A \vee B$

6. $A \equiv B \equiv (-A \vee B) \wedge (-B \vee A)$

2. Coborîm negaţia pe atomi, utilizând echivalenţele:

7. $-A \equiv A$
8. $-(A \wedge B) \equiv -A \vee -B$
9. $-(A \vee B) \equiv -A \wedge -B$ legile lui De Morgan
10. $-\forall x A(x) \equiv \exists x -A(x)$
11. $-\exists x A(x) \equiv \forall x -A(x)$ legile negării cuantificatorilor

3. Redenumim variabilele ce leagă cuantificatori diferiţi utilizând echivalenţele:

12. $Q_1 x P(x) * Q_2 x R(x) \equiv Q_1 x P(x) * Q_2 y R(y)$

unde Q este în 12 o metavariabilă ce denotă pe $\exists$ sau $\forall$, iar * este o metavariabilă ce denotă pe $\wedge$ sau $\vee$)

4. Aplicăm legea extinderii fictive a domeniului unui cuantificator (EFD). Fie A o formulă în care nu apare variabila x. Atunci are loc echivalenţa:

13. $A * Qx P(x) \equiv Qx (A * P(x))$

Formula 13 se foloseşte pentru a deplasa cuantificatorii în faţă. (Prin Qx am notat în 13 din nou pe $\forall x$ sau $\exists x$, iar prin * am notat pe & sau $\vee$). In acelaşi scop se folosesc şi echivalentele:

14. $(\forall x P(x) \wedge \forall x Q(x)) \equiv \forall x (P(x) \wedge Q(x))$
15. $(\exists x P(x) \vee \exists x Q(x)) \equiv \exists x (P(x) \vee Q(x))$

Forma normală prenexă este echivalentă cu formula iniţială de la care am plecat.

Exemplu

16. $\forall x \forall y (V(x,y) \supset \exists t \exists z \exists u (P(z,x) \wedge P(u,y) \wedge P(t,z) \wedge P(t,u))$
17. $\forall x \forall y (-V(x,y) \vee \exists t \exists z \exists u (P(z,x) \wedge P(u,y) \wedge P(t,z) \wedge P(t,u))$
18. $\forall x \forall y \exists t (-V(x,y) \vee \exists z \exists u (P(z,x) \wedge P(u,y) \wedge P(t,z) \wedge P(t,u))$
19. $\forall x \forall y \exists t \exists z \exists u (-V(x,y) \vee (P(z,x) \wedge P(u,y) \wedge P(t,z) \wedge P(t,u))$

Formula 19 este forma normală prenexă a formulei 16.

d. Formele normale Skolem

Dăm acum *regulile de skolemizare*:

Regula S1. Fie A o formulă adusă la forma normală prenexă în care apare un cuantificator existenţial neprecedat de vreun cuantificator universal. Pentru a construi forma normală Skolem procedăm astfel:

1. Eliminăm cuantificatorul existenţial din prefix şi variabila de sub cuantificator:

2. Înlocuim toate aparițiile în matrice (i.e. în corpul formulei) a variabilei de sub cuantificator printr–o constantă individuală inedită, distinctă de constantele care au apărut anterior în calcul.

Regula S2. Fie A o formulă adusă la forma normală prenexă în care apar mai mulți cuantificatori existențiali precedați de cuantificatori universali. Pentru a construi forma normală prenexă procedăm astfel:

1. Eliminăm cuantificatorii existențiali din prefix și variabila de sub cuantificator;

2. Substituim în toate aparițiile sale din matrice variabila de sub cuantificator printr–un termen complex alcătuit dintr–o funcție (cu simbol funcțional inedit) de toți cuantificatorii universali ce preced respectivul cuantificator existențial.

Exemple

20. $\exists x \forall y R(x,y)$

21. $\forall y R(a,y)$

Formula 21 este forma Skolem a formulei 20 și a fost obținută prin aplicarea regulii de skolemizare S1.

19. $\forall x \forall y \exists t \exists z \exists u(-V(x,y) \vee (P(z,x) \wedge P(u,y) \wedge P(t,z) \wedge P(t,u))$

22. $\forall x \forall y(-V(x,y) \vee (P(f(x,y),x) \wedge P(g(x,y),y)) \wedge P(h(x,y),z) \wedge P(h1(x,y),u))$

Formula 22 este forma normală Skolem a formulei 19 și s–a obținut prin aplicarea regulii de skolemizare S2 la 19.

e. Forme normale Davis-Putnam

Forma normală Davis–Putnam este o formă normală în care au fost eliminați cuantificatorii universali din prefix, iar matricea a fost adusă la formă normală conjunctivă. Forma normală Davis Putnam a formulei 16 de mai sus este formula:

23. $(-V(x,y) \vee P(f(x,y),x)) \wedge (-V(x,y) \vee P(g(x,y),y) \wedge (-V(x,y) \vee P(h(x,y),z)) \wedge ((-V(x,y) \vee P(h1(x,y),u))$

și respectiv clauzele:

23a. $-V(x,y) \vee P(f(x,y),x)$ 23c. $-V(x,y) \vee P(h(x,y),z))$

23b. $-V(x,y) \vee P(g(x,y),y)$ 23d. $-V(x,y) \vee P(h1(x,y),u))$

Clauzele 23a – 23d sunt clauze Horn pozitive.

f. Substituția în logica predicatelor

Substituția este o operație prin care într–o formulă bine formată înlocuim o variabilă individuală printr–un termen și obținem, de

asemenea, o formulă bine formată. In anumite circumstanţe putem face abstracţie de expresia în care se face substituţia şi de rezultatul acesteia şi să privim substituţia ca o secvenţă $s = \{ v_1/t_1,...,v_n/t_n \}$ unde v_i sunt variabile şi t_i sunt termeni. Demn de reţinut este faptul că termenii sau *expresiile substituitoare trebuie să nu conţină niciuna dintre variabilele substituite*. Variabilele substituite se mai numesc şi variabile legate (A nu se confunda termenul de variabilă legată în această accepţie cu termenul de variabilă legată în teoria cuantificării).

Numim substituţie vidă o substituţie care nu are nici o pereche variabilă/termen. Substituţia vidă se notează prin ε.

Dacă F este o formulă bine formată în logica predicatelor şi s este o substituţie, atunci Fs va desemna *rezultatul aplicării substituţiei s la formula F* sau o *instanţiere a formulei F prin substituţia s.*

Dacă toţi termenii unei substituţii s sunt constante individuale sau funcţii de constante individuale, atunci F va fi o *instanţiere de bază.* Aceasta va juca rolul furnizării unui exemplu concret sau al coborârii la concret.

Fie s1 şi s2 două substituţii. Spunem că substituţia *s2 este distinctă de s1*, dacă nici o variabilă legată din s1 nu apare în perechile lui s2, dar variabilele legate din s2 pot apare în termenii sau expresiile substituitoare din s1.

Exemplul 1.

$s1 = \{ u/g(x,y) \}$

$s2 = \{ x/a,\ y/b,\ z/f(a,b) \}$

s1 are o singură variabilă legată, pe u, care nu apare în s2. În schimb x şi y, care sunt variabile legate în s2, apar în termenul g(x,y) din s1. s2 este distinctă de s1.

Fie s2 o substituţie distinctă de s1. Vom numi *compunere a lui s2 cu s1* şi scriem $s1^{\circ}s2$ sau mai simplu s1s2 substituţia obţinută prin aplicarea lui s2 la termenii lui s1 şi prin adăugarea la s1 a perechilor variabilă/termen din s2.

Fie s1 şi s2 definite ca în exemplu 1. Atunci compunerea lui s2 cu s1 va fi:

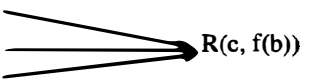
$s1os2 = \{u/g(x,y)\} o \{ x/a,\ y/b,\ z/f(a,b)\} = \{u/g(a,b),\ x/a,\ y/a,\ y/b,\ z/f(a,b)\}$

Compunerea de substituţii este asociativă (vezi demonstraţia în cartea noastră *Logica predicatelor, Hyperion 1992, pp273-275),* dar nu este comutativă.

Fiind dat un set de formule F1,...,Fn şi o substituţie s vom spune că s este o *substituţie unificatoare* pentru setul dat, dacă şi numai dacă, aplicarea lui s la F1,...,Fn duce la aceiaşi formulă, adică F1s = F2s = ...=Fns

Exemplul 2

F1 = R(y, f(z)) {y/c, z/b}
F2 = R(c,u) {u/f(b)}
F3 = B(v, f(b)) {v/c}



Substituţia s = [y/c, z/b, u/f(b), v/c] este unificatoare pentru F1, F2, F3.

De regulă, pentru două formule pot exista mai multe substituţii unificatoare. În aplicarea metodei rezoluţiei suntem interesaţi în a găsi cea mai generală substituţie unificatoare.

Spunem că o substituţie s2 este *tot atât de generală sau mai generală* decât o substituţie s1, dacă şi numai dacă există o altă substituţie s3 care compusă cu s2 să genereze pe s1.

Numim *cel mai general unificator* (mgu i.e. *the most general unifier*) pentru două formule atomare F1 şi F2 o substituţie g care satisface proprietatea ca în raport cu orice substituţie unificatoare s să existe o altă substituţie unificatoare t care compusă cu g şi aplicată la F1 să conducă la acelaşi rezultat ca s aplicată la F1 şi F2.

Exemplul 3

E1 = Q(x,f(y))
E2 = Q(v, f(u))
s1 = { x/c, y/a, v/c, u/a}
s2 = {x/v, y/u}

Este uşor de văzut că s2 este cea mai generală substituţie unificatoare pentru formulele E1 şi E2. Prin aplicarea la E1 a lui s2, E1 devine identic cu E2.

E1s2 = E2 = Q (v, f(u))
E1s1 = Q (c, f (a))
E2s1 = Q (c, f (a))

s1 este o substituţie unificatoare pentru E1 şi E2, dar nu este cea mai generală substituţie unificatoare. Aceasta este s2. Există o substituţie t = { v/c, u/a} care compusă cu s2 generează instanţierea provocată de s1.

E1s2t = E1s1.

A aplica asupra lui E1 compunerea s2t are acelaşi efect ca a aplica asupra lui E1 substituţia s1.

De la cea mai generală substituţie unificatoare printr–o substituţie suplimentară putem întotdeauna obţine rezultatul oricărei alte substituţii unificatoare.

Cea mai generală substituţie unificatoare *evită riscul concretizării abuzive*, căderii nepermise în particular şi prin aceasta pierderii şanselor obţinerii anumitor consecinţe logice.

În literatura de specialitate sunt cunoscuţi mai mulţi algoritmi de unificare: algoritmul de bază, algoritmul lui Ramsay, algoritmul lui Chung–Lee, ecuaţii pe arbori etc.

Doi literali *unifică,* dacă şi numai dacă: 1.*au acelaşi simbol predicativ*; 2. au *acelaşi număr de argumente*; 3. *fiecare dintre argumentele lor având aceiaşi poziţie unifică.*

Două constante individuale distincte nu unifică.

Dimpotrivă, *o pereche variabilă individuală termen aflate pe aceiaşi poziţie în doi literali distincţi, unifică.*

Reprezentările prin arbori sunt indicate pentru a vizualiza procesul unificării.

g. Algoritmi de unificare

Fiind daţi doi literali găsirea substituţiei unificatoare este, de obicei, uşor de identificat. Dar pot fi considerate mulţimi de literali pentru care nu găsim „dintr-o ochire" substituţia unificatoare. În aceste cazuri este nevoie să utilizăm proceduri sau algoritmi speciali de unificare. Expunem mai jos câteva astfel de tehnici formale de unificare.

g1. Algoritmul de unificare de bază

Alain Ramsay prezintă în *Formal Methods in Artificial Intelligence* (Cambridge Univ. Press, 1988) următorul algoritm:

Pentru a unifica *X* şi *Y* procedăm astfel:

(1) Dacă X este o variabilă, observă, mai întâi, dacă acesta a fost anterior unificat cu ceva.

(1.1) Dacă acesta este cazul, atunci îi dăm lui *X* valoarea cu care a fost unificat anterior.

(1.2) În caz contrar, îi dăm acum lui *X* drept valoare pe *Y* sau, altfel exprimat, îl unificăm pe *X* cu *Y*. (Se mai zice că îl „legăm" pe *X* cu *Y*).

(2) Dacă *Y* este o variabilă, observăm mai întâi dacă acesta a fost anterior unificat cu ceva.

(2.1) Dacă acesta este cazul, atunci îi dăm lui *Y* valoarea cu care a fost unificat anterior.

(2.2) În caz contrar, îi dăm acum lui *Y* drept valoare pe *X*.

(3) Dacă *X* şi *Y* sunt ambele expresii complexe, atunci verificăm dacă ele au acelaşi semn predicativ.

(3.1) Dacă au acelaşi semn predicativ, atunci verificăm dacă argumentele lor se unifică.

(3.2) În caz contrar, unificarea eşuează.

(4) Dacă *X* şi *Y* sunt ambele constante, verificăm dacă ele sunt identice. În caz contrar, unificarea eşuează.

(5) Nu sunt alte posibilităţi, căci două expresii sunt sau amândouă expresii complexe (vezi pasul (3)) sau sunt amândouă constante sau, inevitabil, una sau alta dintre ele este o variabilă.

Este uşor de văzut că simbolurile *X* şi *Y* din algoritm ţin loc unor variabile individuale, unor constante individuale sau unor expresii complexe sau literali.

Variabilele sunt interpretate ca parcurgând structuri de date. Când o variabilă nu este interpretată i se asociază eticheta UNDEF; când este interpretată sau valorizată, atunci indicatorul ei se fixează în dreptul expresiei ce reprezintă valoarea ei.

Exemplul 1

Să se unifice expresiile *P(x, f(a), z)* şi *P(b, y, g(u))*.

Evident, nu se aplică instrucţiunile (1) şi (2) pentru că, nici prima, nici a doua expresie nu sunt variabile. Se aplică, în schimb, instrucţiunea (3), căci *P(x, f(a), z)* şi *P(b, y, g(u))* sunt ambele expresii complexe cu acelaşi semn predicativ. În consecinţă, trebuie să verificăm dacă argumentele lor unifică. Pentru aceasta vom verifica pe rând dacă unifică perechile:

$$\left\{\begin{pmatrix} x \\ b \end{pmatrix}\begin{pmatrix} f(a) \\ y \end{pmatrix}\begin{pmatrix} z \\ g(u) \end{pmatrix}\right\}$$

Potrivit instrucţiunii (1) îi dăm lui x valoarea b i.e. x = b, căci x n-a fost anterior astfel valorizat.

Potrivit instrucţiunii (2) îi dăm lui y valoarea f(a), căci şi y este o variabilă ce n-a fost anterior valorizată.

La fel procedăm şi în cazul ultimei perechi $\begin{pmatrix} z \\ g(u) \end{pmatrix}$.

Potrivit instrucţiunii (1), mai exact lui (1.2), cum z n-a fost anterior valorizat, vom postula z = g(u). În consecinţă, substituţia unificatoare este: x = b, y = f(a), z = g(u) sau altfel scrisă s = {x/b, y/f(a), z/g(u)}.

Exemplul 2

Să se unifice $P(a, x, x)$ şi $P(y, y, b)$.

Admitem că iniţial atât x cât şi y sunt nevalorizate. Postulăm, deci, x = UNDEF şi y = UNDEF.

(i) Examinând cele două expresii constatăm că ambele sunt complexe şi au acelaşi semn predicativ. Prin urmare, potrivit instrucţiunii (3) trebuie să verificăm dacă argumentele lor unifică sau nu.

(ii) Prima pereche de argumente este , a este o constantă, iar y este o variabilă care n-a fost anterior valorizată sau unificată cu altceva. Potrivit lui (2.2) unificăm pe y cu a, i.e. scriem $y = a$. Obţinem astfel x = UNDEF, $y = a$.

(iii) A doua pereche de argumente este x şi y, unde x este o variabilă care n-a fost încă valorizată. Potrivit pasului (1.2) legăm pe x cu y.

Obţinem astfel $x = y$ şi $y = a$.

(iv) Ultima pereche este $\begin{pmatrix} x \\ b \end{pmatrix}$. x este o variabilă care a fost anterior valorizată şi, deci, conform pasului (1.1) îi vom asocia lui x exact expresia (valoarea) care i-a fost anterior atribuită, anume y şi vom controla apoi dacă aceasta unifică cu b. Aceasta ne face să parcurgem încă o dată bucla, dar de data aceasta şi y a fost anterior unificat cu a şi va trebui, deci, să ia valoarea a. Trebuie, deci să verificăm dacă a unifică cu b. Dar a şi b sunt constante diferite şi, potrivit pasului (4), ele nu unifică. Deoarece o pereche de argumente $\begin{pmatrix} a \\ b \end{pmatrix}$ nu unifică, expresiile iniţiale, în întregul lor, nu unifică. Algoritmul va afişa eşecul tentativei de unificare.

Este uşor de observat că algoritmul prezentat mai sus este perfectibil. Instrucţiunile 1 şi 2 sunt similare şi pot fi generalizate. Pe acest temei, Alain Ramsay a propus o formă îmbunătăţită a acestuia.

g2. Algoritm de unificare Ramsay îmbunătăţit

Pentru a unifica X şi Y procedăm astfel:

(0) Atât timp cât X şi Y sunt variabile care au fost anterior unificate cu altceva, le vom înlocui cu expresiile cu care au fost unificate (legate).

(1) Dacă X este o variabilă, este una care n-a fost anterior unificată cu nimic altceva (altfel în pasul (0), ea ar fi fost substituită prin expresia cu care a fost unificată anterior). Unifică, deci, acum, pe X cu Y.

(2) Dacă Y este o variabilă, este una care n-a fost anterior unificată cu nimic altceva (altfel în pasul (0), ea ar fi fost substituită prin expresia cu care a fost unificată anterior). Unifică, deci, acum, pe Y cu X.

(3) Dacă X şi Y sunt ambele expresii complexe, verifică dacă ele au acelaşi semn predicativ.

(3.1) Dacă au acelaşi semn predicativ, verifică dacă argumentele lor unifică.

(3.2) Dacă argumentele lor nu unifică, atunci unificarea eşuează.

(4) Dacă X şi Y sunt ambele constante, verifică dacă acestea sunt identice. Altfel, unificarea eşuează.

(5) Nu există alte posibilităţi. Sau cele două expresii sunt ambele complexe sau sunt ambele constante sau cel puţin una dintre ele este o variabilă.

Alain Ramsay arată că un astfel de algoritm rezolvă cu acurateţe unificări de forma $P(x, y, g(x, a))$, $P(f(b), c, u)$ prin $x = f(b)$, $y = c$, $u = g(x, a)$, dar întâmpină dificultăţi în faţa unor expresii de forma $P(x, x)$ şi $P(y, g(y))$. Într-adevăr, aplicarea algoritmului conduce la unificarea $x = y$, $x = g(y)$, ceea se reduce la $y = g(y)$.

Aceasta se întâmplă deoarece prin intermediul lui y şi $g(y)$ se ajunge ca x să fie înlocuit printr-o expresie $g(x)$, ceea ce încalcă definiţia substituţiei (vezi 7.3.2), în particular cerinţa ca *expresiile substituitoare să nu conţină variabilele substituite*. Pentru a preveni acest neajuns ar trebui ca algoritmul nostru să insereze o instrucţiune de verificare a apariţiilor anterioare ale unei variabile în expresiile complexe supuse testului de unificare. Aceasta, însă, ar încetini considerabil procesul de decizie.

g3. Unificarea prin ecuaţiile termenilor

A. Martelli şi U. Montanari au pus la punct, în 1982, un algoritm eficient întemeiat pe definirea unor ecuaţii ale termenilor în care verificarea unificărilor anterioare (occur check) este mult mai lesnicioasă.

Unificarea a doi termeni este, în mare, interpretată ca rezolvare simultană a setului de ecuaţii asociate argumentelor termenilor în cauză. Dacă setul de ecuaţii astfel obţinut are o soluţie, atunci termenii sunt unificabili. Dimpotrivă, dacă setul de ecuaţii obţinut este insolubil, atunci termenii nu unifică.

Adoptând algoritmul celor doi italieni la modul de abordare specific algoritmului de bază îmbunătăţit Alain Ramsay propune, mai întâi, un algoritm de alcătuire a ecuaţiilor şi apoi un algoritm de procesare a acestora.

c1. *Algoritmul de alcătuire a ecuaţiilor* este următorul:

Pentru a crea un set de ecuaţii în vederea unificării lui X şi Y, procedăm astfel:

(1) Dacă X este o variabilă, adaugă $X = Y$ şi reia (return).

(2) Dacă Y este o variabilă, adaugă $Y = X$ şi reia.

(3) Dacă X şi Y sunt constante, verifică dacă ele sunt identice. Dacă sunt, atunci reia. Altfel, unificarea eşuează.

(4) Dacă X şi Y sunt expresii complexe, verifică dacă ele au acelaşi semn predicativ. Dacă au acelaşi semn predicativ, formulează ecuaţiile corespunzătoare pentru perechile lor de argumente şi reia. În caz contrar, unificarea eşuează.

(5) Eşec se poate obţine numai dacă una dintre X ori Y este o constantă, iar cealaltă este o expresie complexă, caz în care apare o problemă.

c2. *Prelucrarea setului de ecuaţii* se desfăşoară astfel:

(1) Pentru orice ecuaţii de forma $X = Y$, unde atât X cât şi Y sunt variabile, adaugă Y în partea stângă a oricărei alte ecuaţii, având pe X în partea stângă.

(2) Unifică (merge) orice pereche de ecuaţii care au aceeaşi variabilă în partea stângă a ecuaţiei.

Pentru a unifica două ecuaţii procedăm astfel:

(1) Creează o nouă ecuaţie prin unificarea părţilor stângi şi prin unificarea părţilor din dreapta a două ecuaţii anterioare.

(2) Înlătură ecuaţiile iniţiale din setul de ecuaţii.

(3) Realizează algoritmul c1 pentru orice pereche de componente (items) din cele două părţi din dreapta (i.e. pentru primul element din prima parte din dreapta asociat cu fiecare element din cea de a doua parte din dreapta, apoi pentru cel de-al doilea element din prima parte din dreapta, asociat la fel cu fiecare dintre elementele din cea de a doua parte dreaptă şi aşa mai departe).

Exemplul 1. Să considerăm din nou $P(a, x, x)$ şi $P(y, y, b)$ ca cei doi termeni a căror unificare vrem să o testăm prin metoda ecuaţiilor.

Potrivit instrucțiunii (4) din c1 scriem ecuațiile: $y=a$, $x=y$, $x=b$. Considerăm primele două ecuații $y=a$, $x=y$. Întrucât și x și y sunt variabile în $x=y$, și y este partea stângă a ecuației $y=a$, potrivit instrucțiunii (1) din c2 scriem $|y, x|=a$ alături de cea de a treia ecuație inițială $x=b$. Potrivit instrucțiunii (2) din c2, din $|y, x|=a$ și $x=b$ rezultă $|y, x|=|a, b|$. Dar $|a, b|$ este în sine inconsistent, căci a și b sunt constante diferite ce nu unifică.

Concluzia este cea firească și anume că $P(a, x, x)$ și $P(y, y, b)$ nu unifică.

g4. Ecuații pe arbori

Prologul este un limbaj ce manipulează arbori. Nu este, deci, de mirare că și problema unificării unor literali este abordată tot din perspectiva teoriei formale a arborilor.

Mai întâi, vom arăta că oricăror literali, în această metodă, li se asociază arbori. Fie $P(x, f(x), b)$ și $P(a, y, z)$ doi literali distincți.

Lor le asociem arbori după următoarea regulă. Semnul predicativ va eticheta rădăcina arborelui, argumentelor predicatului le vom asocia noduri imediat descendente din rădăcină. Se înțelege, numărul nodurilor descendente coincide cu $\delta(P)$ sau aritatea predicatului în cauză.

Aplicând această regulă, celor doi literali de mai sus le vom asocia arborii:

Doi literali unifică, dacă și numai dacă ecuațiile lor definite pe arbori au o soluție. Soluția unui astfel de sistem, dacă există, va consta dintr-o mulțime de valori date variabilelor.

Ecuația sistemului asociat celor doi literali de mai sus este $x=a$, $y=f(x)$, $z=b$, ceea ce devine imediat $s= [x/a, y/f(x), z/b]$.

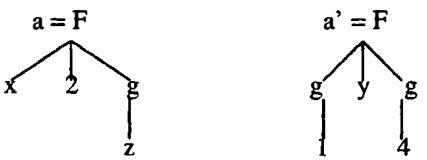
În aceste sisteme de ecuații intervin constrângeri elementare alcătuite din egalități sau inegalități între arbori, de forma $a=a'$ $a \neq a'$, unde a și a' sunt arbori.

Constrângerile unui sistem de ecuații sunt compuneri de constrângeri elementare.

Interpretorul Prolog utilizează un algoritm de reducere a constrângerilor complexe la constrângeri elementare sau ireductibile. Vor fi *constrângeri reduse* cele care 1) conţin o egalitate între doi arbori care nu au aceeaşi rădăcină sau dacă au aceeaşi rădăcină 2) nu au acelaşi număr de argumente, cazuri în care sistemul iniţial nu are soluţie şi, în sfârşit, 3) conţin doar egalităţi având ca membru stâng variabile distincte, iar ca membru drept termeni elementari sau compuşi, caz în care constrângerile curente indică soluţia sistemului.

A rezolva un sistem de ecuaţii de acest tip înseamnă a trece succesiv de la constrângeri complexe vizând arbori la constrângeri reduse.

Exemplul 1



Fie $a=F(x, 2, g(z))$ $a'=F(g(1), y, g(4))$

Scriem secvenţa de constrângeri $C_0,...,C_n$

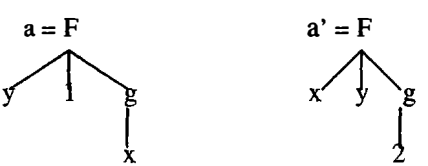
$C_0=\{F(x, 2, g(z)) = F(g(1), y, g(4))\}$

$C_1=\{x=g(1), y=2, g(z)=g(4)\}$

$C_2=\{x=g(1), y=2, z=4\}$.

C_2 reprezintă soluţia sistemului de ecuaţii şi în acelaşi timp substituţia unificatoare care poate fi scrisă ca $mgu=\{x/g(1), y/2, z/4\}$.

Exemplul 2



Fie $a=F(y, 1, g(x))$ $a'=F(x, y, g(2))$

$C_0=\{F(y, 1, g(x))=F(x, y, g(2))\}$

$C_1=\{y=x, y=1, g(x)=g(2)\}$

$C_2=\{y=x, y=1, x=2\}$

$C_3=\{y=x, x=1, x=2\}$

$C_4=\{y=x, y=1, 1=2\}$.

Este evident, că cei doi literali nu unifică, deoarece ecuaţiile asociate lor, nu au o soluţie consistentă. Din C_3 x=1 şi x=2 rezultă un neadevăr matematic exprimat în C_4 prin 1=2.

Este uşor de remarcat o apropiere între metoda ecuaţiilor pe arbori prezentată mai sus şi unificarea prin ecuaţii ale termenilor prezentată anterior la c_1 şi c_2.

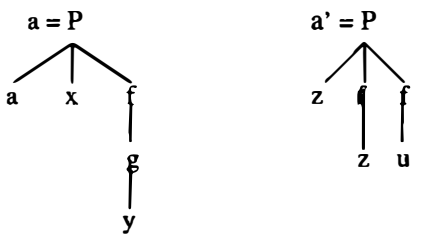
În starea C_3 putem lesne aplica instrucţiunile (1) şi (2) din c_2 şi să obţinem:

1) $\{x, y\}=1$ şi $x=2$ (prin (1) din c_2)

2) $\{x, y\}=\{1, 2\}$ (prin (2) din c_2)

şi din nou să ne oprim pentru că $1 \neq 2$.

Exemplul 3



$a=P(a, x, f(g(y)))$ $a'=P(z, f(z), f(u))$

Secvenţa de constrângeri va fi:

$C_0=\{P(a, x, f(g(y)))=P(z, f(z), f(u))\}$

$C_1=\{z=a, x=f(z), f(g(y))=f(u)\}$

$C_2=\{z=a, x=f(z), u=g(y)\}$

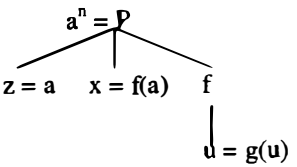
$C_3=\{z=a, x=f(a), u=g(y)\}$

C_3 este redusa egalităţii dintre arborii a şi a' şi reprezintă soluţia sistemului de ecuaţii formulat la C_1 şi C_2.

Substituţia unificatoare este, deci, mgu=$\{z/a, x/f(a), u/g(y)\}$.

Ulterior vom testa aceeaşi unificare propusă în exemplul 3 printr-un alt algoritm de unificare.

Este lesne de observat că soluţia găsită în C_3 ne sugerează un alt arbore, de la care pornind putem lesne genera pe *a* şi pe *a'*, prin utilizarea soluţiilor găsite pentru sistemul de ecuaţii asociate arborilor.



Arborele corespunzând celui mai general unificator mgu *este un subarbore al celor doi arbori daţi iniţial*, de la care pornind, prin valorizări suplimentare, putem reconstitui arborii daţi iniţial.

Exigenţa mgu este de a nu ne angaja niciodată la o interpretare particulară decât cea strict necesară, pentru a ne lăsa deschise cât mai multe posibilităţi de manevră sintactico-semantică.

g5. Algoritmul Chung şi Lee

În *Symbolic Logic and Mathematical Theorem Proving* Chin Liang Chang şi Richard Char Tung Lee dau un algoritm de unificare întemeiat pe ideea de diferenţă sau dezacord existent între doi sau mai mulţi literali. Aşa, de exemplu, între literalii *P(x)* şi *P(a)* mulţimea de dezacord constă în $\{x, a\}$, unde *x* este o variabilă, iar *a* este constantă. Pentru ca dezacordul să dispară şi cei doi literali să unifice, cum *x* este o variabilă, este suficient să substituim pe *x* cu *a*, i.e. să stipulăm substituţia $s=\{a/x\}$ pentru cei doi literali. Algoritmul Chung-Lee este o generalizare şi sistematizare a acestei observaţii.

Definim mai întâi mulţimea de dezacord a doi sau mai mulţi literali.

Fie *E* o mulţime de literali. Numim mulţime de dezacord o mulţime *E* de termeni diferiţi având aceeaşi poziţie ca argumente în literalii din *E* (numărătoarea argumentelor se face de la stânga la dreapta).

Exemplu. Fie E=P(x, f(x, y)), P(x, a), P(x, g(h(k(x))))

Mulţimea de dezacord pentru E este mulţimea $E^d=\{f(x, y), a, g(h(k(x))))\}$.

Fie W o mulţime de literali, W_0 mulţimea iniţială de literali, W_k mulţimea curentă, σ_k substituţia curentă, ε substituţia vidă.

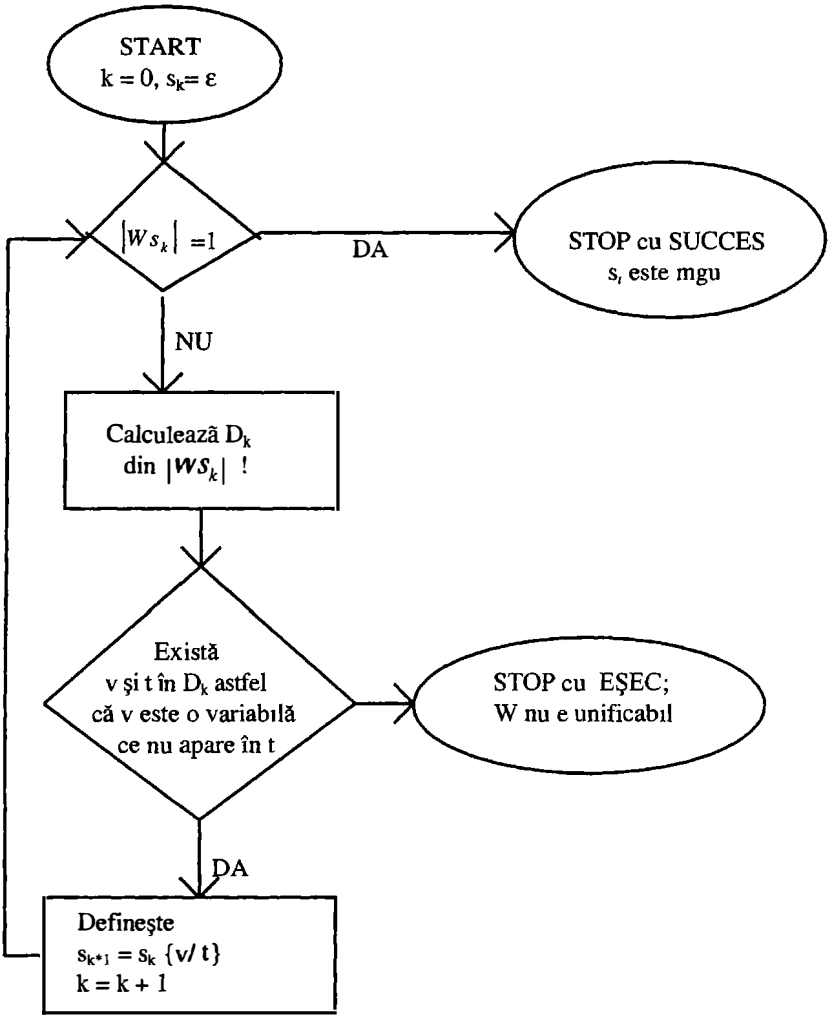


Fig V.1 Organigrama ciclului de bază al algoritmului de unificare

Atunci, *algoritmul lui Chang şi Lee* poate fi formulat astfel:

Pasul 1. Pune k=0, W_k=W şi s_k=ε.

Pasul 2. Dacă W_k este o mulţime singleton (cu un singur element), atunci stop; s_k e cel mai general unificator, i.e. s_k este un mgu pentru W.

Altfel caută mulţimea de dezacord D_k a lui W_k.

Pasul 3. Dacă există elemente v_k şi t_k în D_k astfel că v_k este o variabilă care nu apare în t_k, atunci du-te la Pasul 4. Altfel stop; W nu este unificabilă.

Pasul 4. Fie $s_{k+1}=s_k\{t_k/v_k\}$ şi $W_{k+1}=W_k\{t_k/v_k\}$ (De reţinut că $W_{k+1}=Ws_{k+1}$)

Pasul 5. Pune k=k+1 şi du-te la Pasul 2.

Observaţii

Pasul 1 iniţializează procesul. *W* este setul iniţial de literali, *k*=0 şi substituţia iniţială este substituţia vidă, fără nici o pereche termen/variabilă. Cei doi autori scriu substituţia sub forma termen/variabilă nu sub forma variabilă/termen.

Dacă W_k este o mulţime singleton, atunci unificarea s-a realizat iar substituţia s_k este cel mai general unificator, i.e. mgu. În caz contrar, se caută mulţimea de dezacord a mulţimii W_k, notată prin D_k.

Paşii 3 şi 4 ne indică ce substituţii suplimentare trebuie să facem, (ei corespund formării de noi ecuaţii) şi cum să calculăm rezultatul substituirii.

Pasul 5 asigură ciclarea procesului punând $k=k+1$ şi trimiţând la pasul 2.

Funcţionarea algoritmului este redată în organigrama de mai jos:

Exemplul 1. Să se găsească cel mai general unificator pentru $W=P(a, x, f(g(y))), P(z, f(z), f(u))$.

1. Postulăm $s_0=\varepsilon$ şi $W_0=W$. Deoarece W_0 nu este o mulţime singleton s_0 nu este cel mai general unificator pentru *W*.

2. Trebuie, deci, să calculăm mulţimea de dezacord pentru W_0, i.e. mulţimea $D_0=\{a, z\}$. În D_0 există o variabilă $v_0=z$, care nu apare în termenul t_0, $t_0=a$.

3. Pe acest temei trecem la pasul 4 şi definim substituţia s_1 şi mulţimea W_1.

$s_1=s_0 \circ \{t_0/v_0\}=\varepsilon \circ \{a/z\}=\{a/z\}$

$W_1=W_0\{t_0/v_0\}$

$=\{P(a, x, f(g(y))), P(z, f(z), f(u))\}\{a/z\}$

$=\{P(a, x, f(g(y))), P(a, f(a), f(u))\}$

Considerăm acum drept pas curent pe 1, conform Pasului 5 şi trecem la Pasul 2.

4. Constatăm că W_1 nu este o mulţime singleton şi, în consecinţă, îi calculăm mulţimea de dezacord, care corespunde acum celui de al doilea argument din cei doi literali din W_1.

$D_1=\{x, f(a)\}$

5. În D_1 găsim $v_1=x$ şi $t_1=f(a)$ astfel că v_1 nu apare în t_1 şi deci, putem trece la Pasul 4 şi să definim:

$$s_2 = s_1 \circ \{t_1/v_1\}=\{a/z\}\circ\{f(a)/x\}=\{a/z, f(a)/x\}$$

$$W_2 = W_1\{t_1/v_1\}$$

$$=\{P(a, \lambda, f(g(y))), P(a, f(a), f(u))\}\{f(a)/x\}$$

$$=\{P(a, f(a), f(g(y))), P(a, f(a), f(u))\}$$

6. Prin Pasul 5 considerăm $k=2$ şi trecem din nou la Pasul 2, unde constatăm că nici W_2 nu este o mulţime singleton. În consecinţă, calculăm mulţimea de dezacord pentru W_2:

$D_2=\{g(y), u\}$

În D_2 identificăm variabila $v_2=u$ şi termenul $t_2=g(y)$. Cum însă variabila u nu apare în $g(y)$, putem adăuga substituţia

$$s_3 = s_2 \circ \{t_2/v_2\}=\{a/z, f(a)/x\}\circ\{g(y)/u\}$$

$$=\{a/z, f(a)/x, g(y)/u\}$$

şi evident

$$W_3 = W_2\{t_2/v_2\}$$

$$=\{P(a, f(a), f(g(y)), P(a, f(a), f(u))\}\{g(y)/u\}$$

$$=\{P(a, f(a), f(g(y)), P(a, f(a), f(g(y)))\}$$

$$=\{P(a, f(a), f(g(y))\}.$$

7. Deoarece W_3 este o mulţime singleton, s_3 este cel mai general unificator pentru mulţimea iniţială W.

g6. Observaţii finale

Am prezentat anterior câteva proceduri formale pentru identificarea celei mai generale substituţii unificatoare. De la aceasta pornind, putem afla lesne orice altă substituţie unificatoare.

După ce criterii exterioare putem identifica literalii sau expresiile care unifică de cele care nu unifică.

Este evident că două constante diferite nu unifică niciodată. La fel nu unifică niciodată două formule compuse ce au functori (i.e. semne funcţionale sau semne predicative) diferiţi(e). Chiar având acelaşi functor, două expresii pot să nu unifice, dacă au numere de argumente diferite. Toate acestea sunt cazuri de eşec, pe care algoritmii prezentaţi mai sus le vor semnala ca eşecuri.

Dimpotrivă, pentru ca un algoritm să ne dea pentru două expresii x şi y drept răspuns cea mai generală substituţie unificatoare mgu(x, y) este necesar ca cele două condiţii de mai sus să fie satisfăcute (x şi y au acelaşi functor, au acelaşi număr de argumente) şi în plus argumentele lor trebuie să unifice şi să genereze o ecuaţie solubilă.

Am văzut anterior că, fiind date două expresii (de exemplu doi literali) E_1 şi E_2 şi o substituţie unificatoare a lor s, dacă γ este cel mai general unificator al lui E_1 şi E_2, atunci există întotdeauna o substituţie σ care compusă cu γ să satisfacă condiţia:

$E_1 \gamma \sigma = E_1 s = E_2 s$.

Cea mai generală substituţie unificatoare ne garantează menţinerea noastră la un nivel minim de interpretare sau valorizare, fără să ne blocăm pretimpuriu posibilităţile de interpretare.

Este important de menţionat faptul că algoritmii de unificare sunt captaţi în procedurile formale ale sistemelor de calcul, utilizatorul fiind astfel eliberat de grija căutării individuale, de la caz la caz, a celei mai generale substituţii unificatoare.

Dăm mai jos o procedură de calcul a celui mai general unificator preluată după M.R. Genesereth şi Nils J. Nilsson:

```
    Recursive Procedure Mgu(x, y)
    Begin x = y ⇒ Return
    Variable(x) ⇒ Return (Mguvar(x, y)),
    Variable(y) ⇒ Return (Mguvar(x, y)),
    Constant (x) or Constant (y) ⇒ Return(False)
    Not (Length (x) = Lenght (y)) ⇒ Return(False),

                g ← [ ]
    Tag         i = Length (x) ⇒ Return (g)
                s ← Mgu(Part(x, i), Part(y,i))
                s = False ⇒ Return (False),
                g ← Compose (g, s),
                x ← Substitute (x, g)
                y ← Substitute (y, g)
                i ← i +1
                Go to Tag
    End
    End
Procedure Mguvar (x, y)
    Begin Includes (x, y) ⇒ Return (False)
Return [x / y]
      End
```

Dacă două expresii sunt unificabile, procedura ne dă cel mai general unificator, altfel întoarce (return) False.

Procedura explorează pe rând alternativa că expresia inițială poate fi o constantă, o variabilă sau un obiect structurat, i. e. o expresie complexă.

Este uşor să regăsim în procedura dată observaţiile noastre de mai sus. Evident, predicatele Variable şi Constant sunt adevărate despre obiectele matematice în cauză. Lenght, lungimea indică numărul de argumente ale literalilor sau semnelor funcţionale numite aici obiecte structurate. Când cele două obiecte structruate x şi y nu au aceiaşi lungime, atunci ele nu unifică şi algoritmul returnează fals.

Un obiect structurat constă dintr-un semn predicativ sau semn funcţional sau un operator şi dintr-un număr de argumente. Semnul predicativ sau semnul funcţional alcătuieşte partea zero a obiectului complex, iar argumentele acestora alcătuiesc cealaltă parte, obiectul în cauză fiind de lungime 2. Dacă x' P (a, g(z)), atunci vom avea Part (x, 1) = *a*, Part (x, 2) = g(y).

În procedura de mai sus apar unele subrutine nedefinite aici. De exemplu, *Substituţie* are ca argumente o expresie şi o substituţie (i.e. o mulţime de perechi termeni/variabile) şi întoarce ca răspuns rezultatul substituţiei. Mai departe, *Compose* ia ca argumente două substituţii şi întoarce ca rezultat compunerea lor. În sfârşit, *Includes* ia ca argumente o variabilă şi o expresie şi întoarce True, dacă şi numai dacă variabila apare în expresie. *Includes* este utilizat pentru a verifica apariţia unei variabile într-o expresie anterioară (occur check).

Putem acum trece la formularea *principiului rezoluţiei.*

h. Principiul rezoluţiei şi teorema rezolventului în logica predicatelor

Fiind date două clauze k1 şi k2, astfel că $\ell_1 \in$ k1, $\ell_2 \in$ k2 şi s este cea mai generală substituţie unificatoare pentru ℓ_1 şi ℓ_2, putem conchide asupra clauzei rezolvent obţinută prin aplicarea substituţiei s la mulţimea literalilor din k1 şi din k2 după ce s–a eliminat perechea opusă de literali ℓ_1 şi ℓ_2.

Fiind date două clauze adevărate, k1 şi k2, ce conţin o pereche opusă de literali ℓ_1 şi ℓ_2 astfel că $\ell_1 \in$ k1, $\ell_2 \in$ k2 şi având drept substituţie unificatoare pe s astfel că $\ell_1 s = \ell_2 s$, putem conchide asupra adevărului clauzei rezolvent obţinută prin aplicarea substituţiei s la mulţimea literalilor din k1 şi din k2, din care s-a eliminat perechea opusă de literali.

Schematic aceasta poate fi redată astfel:

ℓ_1	ℓ_2	
k1	k2	$\ell_1 s = \ell_2 s$

$res_r(k1, k2) = (k1 \setminus \ell_1 \cup k2 \setminus \ell_2) s$

Inferenţa rezolutivă se aplică după ce s–a descoperit cea mai generală substituţie unificatoare.

Perechea opusă de literali se elimină. Rezolventul este mulţimea literalilor din k1 şi din k2 cărora li s-a aplicat substituţia unificatoare s şi dintre care s-a eliminat perechea opusă de literali.

Formularea de mai sus este satisfăcătoare pentru cazurile când în k1 şi k2 există numai o pereche de literali în care se poate face rezoluţia. Pentru cazurile când în k1 şi k2 există mai multe cupluri de literali rezolubili se defineşte clauza factor.

Fiind dată o clauză iniţială k, ce conţine mai multe apariţii ale aceluiaşi literal P şi cea mai generală substituţie unificatoare s pentru apariţiile literalului P în k, vom numi *clauză factor* o clauză k' obţinută prin aplicarea lui s la literalii din clauza iniţială:

$k' = ks$ (CF)

Numim *literal factor* un literal ℓ_1 obţinut prin aplicarea substituţiei mgu = s_1 la literalii iteraţi din clauza iniţială:

$\ell_1 = P_1 s_1$ (LF)

Exemple

k1 = { P(x), P(f(y)), R(x, y)}

k2 = {-P(z), -P(f(u)), S(u, t) }

Este uşor de observat că în k1 apar literalii P(x) şi P(f(y)), iar în k2 apar literalii -P(z) şi -P(f(u)).

Cea mai generală substituţie unificatoare pentru literalii P(x) şi P(f(y)) din k1 este s1 = {x/f(y)}, iar clauza factor a lui lui k1 este:

k1' = {P(f(y)), R(f(y), y)}

Similar, cea mai generală substituţie unificatoare, mgu, pentru literalii -P(z), -P(f(u)) din k2 este s2 = {z/f(u)}iar clauza factor a lui k2 este:

k2' = {-P(f(u)), S(u, t)}.

P(f(y)) este literalul factor în k1', iar -P(f(u)) este literalul factor în k2'

Este uşor de observat că expresiile clauzale k1' şi k2' sunt rezolubile, dacă vom substitui în k2' pe u prin y, i.e. mgu între k1' şi k2' este s3 = {u/y}.

Obţinem astfel:

$res_P(k1', k2') = \{R(f(y), y), S(y, t)\}$

Putem acum să ne întoarcem la formularea principiului rezoluţiei în logica predicatelor, utilizând, de această dată şi conceptul de clauză factor.

Definiţie. Fie k1 şi k2 doua clauze. Daca există un literal ℓ_1 într-un factor k1' al lui k1 şi un literal $-\ell_2$ într-un factor k2' al lui k2, astfel că ℓ_1 şi ℓ_2 au un cel mai general unificator s, atunci vom spune că cele două clauze k1 şi k2 rezolvă şi că rezolventul lor este :

$(k1' \setminus \{\ell_1\} \cup k2' \setminus \{-\ell_2\})s$.

Demersul inferenţial rezolutiv poate fi redat prin schema:

$$\frac{k1 \qquad \ell_1 \in k1' \qquad k2 \qquad -\ell_2 \in k2'}{res_\ell(k1, k2) = (k1' \setminus \{\ell_1\} \cup k2' \setminus \{-\ell_2\})s}$$

Astfel definit principiul rezoluţiei presupune aplicarea de trei ori a algoritmului de unificare, de două ori pentru identificarea clauzelor factor k1' şi k2' şi a treia oră pentru a găsi substituţia unificatoare între ℓ_1 şi $-\ell_2$ din clauzele factor k1' şi k2'

Principiul rezoluţiei ne apare astfel indisolubil legat de algoritmii de unificare, chiar dacă pentru un număr de cazuri substituţiile unificatoare sunt uşor de identificat.

Principiul rezoluţiei, ca şi modus ponens, modus tollens, principiul tranzitivităţii implicaţiei, principiul raţionamentului ipotetic disjunctiv etc. este o schemă validă de inferenţă.

Aceasta se poate vedea din *teorema rezolventului.*

Dacă k1 şi k2 sunt două clauze adevărate şi $\ell \in k1$ şi $-\ell \in k2$, atunci clauza rezolvent $res_\ell(k1, k2)$ va fi şi ea adevărată. În cartea noastră *Logica predicatelor* am prezentat trei modalităţi de demonstrare a acestei teoreme la nivelul logicii propoziţiilor. Dăm, în continuare, demonstrarea acestei teoreme în logica predicatelor de ordinul întâi.

Presupunem asimilate noţiunile de substituţie unificatoare, cea mai generală substituţie unificatoare, clauză factor şi literal factor definite mai sus, precum şi definiţia noţiunii de rezolvent în logica predicatelor.

Simbolic, teorema rezolventului poate fi exprimată astfel:

$k_1 \wedge k_2 \wedge P_i \in k_1 \wedge P_j \in k_2 \vdash res_P(k_1, k_2)$

Demonstraţie.

1.	$k_1 \wedge k_2 \wedge P_i \in k_1 \wedge P_j \in k_2$	ip.
2.	$2 . k_1$	E $\wedge$, 1
3	$3\ k_2$	E $\wedge$, 1
4.	$P_i \in k_1$	E $\wedge$, 1
5.	$P_j \in k_2$	E $\wedge$, 1
6	$mgu(P_i) = s_1$	ip.
7.	$mgu(P_j) = s_2$	ip
8.	$8\ mgu(P_i , P_j) = s_3$	ip.
9.	$9.\ k_1' = k_1 s_1$	D
10.	$10.k_2' = k_2 s_2$	D
11	$11.l_1 = P_i s_1$	D
12.	$l_2 = P_j s_2$	D
13.	$l_1 \in k'_1$	(4, 9, 1)
14	$l_2 \in k'_2$	(5, 10, 12)
15.	$k_1 s_1$	(Rs, 2)
16.	k'_1	(RE 9, 15)
17.	$k_2 s_2$	(Rs,3)
18.	k'_2	(RE 10, 17)
19.	$l_1 = 0$	ip.supl.
20.	$k'_1 \setminus \{l_1\}$	(13, 16, $\vee$)
21.	$(k'_1 \setminus \{l_1\} \cup k'_2 \setminus \{l_2\})$	(20, $\vee$)
22.	$(k'_1 \setminus \{l_1\} \cup k'_2 \setminus \{l_2\})s_3$	(Rs 21)
23.	$res_p (k_1 , k_2)$	(RE, 22, D1)
24.	$l_1 = 1$	ip supl.
25.	$l_2 = 0$	(18, 25, $\vee$)
26.	$k'_2 \setminus \{l_2\}$	(26, $\vee$)
27.	$k'_1 \setminus \{l_1\} \cup k'_2 \setminus \{l_2\}$	
28.	$(k'_1 \setminus \{l_1\} \cup k'_2 \setminus \{l_2\})s_3$	(Rs 27)
29.	$res_p (k_1 , k_2)$	(RE, 28, D1)

Am admis în demonstraţia noastră că ambele clauze conţin iteraţii de literali: k_1 pe $P_1 \ldots P_n$, iar k_2 pe $P_1 \ldots P_m$, cu $1 \leq m,\ n \leq \infty$. În consecinţă, înainte de a aplica principiul rezoluţiei este nevoie să unificăm prin s_1 pe $P_1 \ldots P_n$, iar prin s_2 pe $P_1 \ldots P_m$, ultimii fiind prefixaţi de semnul negaţiei '-'

k'_1 şi k'_2 sunt clauzele factor rezultate prin aplicarea substituţiilor corespunzătoare, iar l_1 şi l_2 sunt literalii factor obţinuţi în acelaşi fel. Cum $l_1 \in k'_1$ şi $l_2 \in k'_2$ restul demonstraţiei este analoagă uneia dintre versiunile prezentate în paragraful 16.2

Singura deosebire constă în aplicarea regulii Rs care susține că substituția uniformă conservă adevărul clauzelor pe temeiul menționat mai sus.

Mediată de substituțiile unificatoare, trecerea de la clauze la rezolvent rămâne o schemă validă de inferență.

Care sunt, aşadar, momentele principale ale aplicării principiului rezoluției la clauzele inițiale k_1 ce conține literalii $P_1 \ldots P_n$, şi k_2 ce conține literalii $P_1 \ldots P_m$? Acestea sunt: **1.** aflarea celei mai generale substituții unificatoare pentru literalii $P_1 \ldots P_n$ din k_1 i.e. mgu $(P_1) = s_1$; **2.** aflarea mgu pentru $P_1 \ldots P_m$ din k_2 i.e. mgu $(P_1) = s_2$; **3.** aflarea clauzei factor din k_1 prin aplicarea lui s_1 la k_1, i.e. $k'_1 = k_1 s_1$ şi aflarea clauzei factor din k_2 i.e. $k'_2 = k_2 s_2$; **4.** aflarea literalilor factor din $P_1 \ldots P_n$ şi din $P_1 \ldots P_m$; **5.** descoperirea celei mai generale substituții unificatoare pentru literalii factor l_1 şi l_2 i.e. descoperirea lui s_3. **6.** utilizarea reguli Rs pentru a ne convinge de adevărul clauzelor k'_1 şi k'_2 După ce deținem pe k'_1 , k'_2 şi după ce ne-am calculat literalii factor l_1 şi l_2 demonstrația se desfăşoară ca în cazul unei versiuni de demonstrare a teoremei rezolventului în logica de propoziției (cu excepția aplicării regulii Rs).

În demonstrația de mai sus au intervenit de trei ori algoritmi de unificare şi tot de atâtea ori instanțieri ale unor formule k_1 k_2 prin substituțiile unificatoare, instanțieri ale lui k'_1 şi k'_2.

Desigur, dacă cele două clauze inițiale nu conțin fiecare aparițiii iterate ale literalilor în apariție pozitivă $P_1 \ldots P_n$, sau ale literalilor $P_1 \ldots P_m$ în apariție negată, atunci nu e nevoie să procedăm la aplicarea algoritmului de unificare pentru respectiva clauză inițială. Putem întâlni şi cazul mai fericit când nici k_1 nici k_2 nu conțin literali iterați. Atunci l_1 se află în k_1 şi l_2 se află în k_2 şi suntem scutiți de a descoperi s_1 şi s_2.

Principiul rezoluției în logica predicatelor presupune utilizarea algoritmilor de unificare. Deşi mai complicat prin utilizarea acestora, el rămâne o regulă validă de inferență, ca şi în logica propozițiilor şi stă la baza metodei rezoluției.

În paragraful 16.3 am definit conceptul de derivare rezolutivă în logica propozițiilor, sprijinindu-ne pe principiul rezoluției. Conceptele de derivare rezolutivă şi de respingere rezolutivă întemeiate pe aplicarea iterată a principiului rezoluției sunt valabile şi în logica predicatelor de ordinul întâi. Mai departe, în 16.4, am demonstrat prin inducție teorema derivării rezolutive. Dacă S este un set adevărat de clauze şi dacă din S derivă rezolutiv o clauză k, atunci din S se deduce

logic *k*. Derivarea rezolutivă devine astfel echivalentă cu deducerea dintr-un set de clauze sau premise date. Este important de arătat aici că noţiunile de derivare rezolutivă sunt valabile şi despre clauzele din logica predicatelor. Unificarea se face la nivelul pasului rezolutiv şi nu împiedică valabilitatea teoremei consecinţei rezolutive la nivelul logicii predicatelor.

Fiind demonstrată teorema rezolventului şi păstrând valabilitatea teoremei consecinţei rezolutive (nu neapărat imediate) şi în logica predicatelor putem, în sfârşit, trece la definirea metodei rezoluţiei în logica predicatelor.

i. Caracterizarea metodei rezoluţiei în logica predicatelor

Metodele sunt mulţimi de reguli sau instrucţiuni cu ajutorul cărora rezolvăm anumite probleme sau trecem de la o stare iniţială în care ne asumăm un scop la o stare finală în care acesta este realizat. Metodele sunt, într-un anumit sens, operaţionalizări de scopuri. Menirea lor este să facă activităţile umane eficace. Conceptul de metodă ţine atât de logică şi de epistemologie, cât şi de praxiologie ca teorie a activităţilor umane eficiente.

Metodele au condiţii de început, condiţii de terminare şi instrucţiuni sau reguli ordonate, precum şi numeroase condiţii şi teste intermediare.

Rezoluţia este o specie de deducţie formală întemeiată pe formele clauzale ce constă în aplicarea repetată a principiului rezoluţiei până ce este atinsă condiţia de terminare.

Metoda rezoluţiei este întotdeauna o derivare rezolutivă sau o respingere rezolutivă. Derivarea rezolutivă corespunde demonstraţiei directe din ipoteze, iar respingerea rezolutivă corespunde demonstraţiei prin reducere la absurd.

Condiţia de început a unei derivări rezolutive constă în setul de clauze provenite din premisele unui raţionament sau din datele iniţiale ale problemei de demonstrat

Condiţia de început a unei respingeri rezolutive constă din clauze obţinute din premise la care se adaugă clauzele obţinute din negaţia concluziei (i. e. negaţia clauzei scop).

Condiţia de terminare a derivării rezolutive constă în atingerea clauzei scop sau a tezei de demonstrat sau în găsirea unui contramodel. Condiţia de terminare a respingerii rezolutive constă în obţinerea clauzei vide, respectiv a unui set incompatibil de clauze.

Derivarea şi respingerea rezolutivă sunt ambele aplicări repetate ale principiului rezoluţiei, care se deosebesc între ele doar prin condiţiile de începere şi terminare.

Particularitatea metodei rezoluţiei ca metodă deductivă constă în utilizarea *unei singure reguli de derivare*, principiul rezoluţiei, pe care o repetă până ce este atinsă condiţia de terminare.

Din acest punct de vedere, metoda rezoluţiei poate fi asemănată cu unele sisteme axiomatice deductive în care se foloseşte, în exclusivitate, o singură regulă de inferenţă, regula modus ponens.

Dar sistemele axiomatice operează cu teoreme cărora le corespund logic-funcţional tautologii, în timp ce metoda rezoluţiei are ca puncte de plecare formule adevărate, nu tautologii.

Rafinarea metodei rezoluţiei va privi cu prioritate criteriile de alegere a perechilor de clauze şi a literalilor în care se face rezoluţia şi, în funcţie de aceştia, a algoritmilor de unificare a literalilor în cauză. Rezoluţia poate fi definită ca o procedură computaţională. Astfel M. R. Genesereth şi Nils J. Nilsson o definesc în *Logical Foundations of Artificial Intelligencei astfel*:

Procedure Resolution (*Delta*)
Repeat Termination (*Delta*) ⇒ *Return* (*Success*),
Phi ← *Choose* (*Delta*), *Psi* ← *Choose* (*Delta*),
Chi ← *Choose* (*Resolvents* (*Phi, Psi*)),
Delta ← *Concatenate* (*Delta, Chi*)
End

Delta este o mulţime de clauze reprezentând baza de cunoştinţe. Primul rând din procedură ne spune că dacă este satisfăcută condiţia de terminare, atunci calculatorul ne raportează succesul demonstraţiei.

În derivarea rezolutivă vom avea:

Termination (*Delta*) = *Goal* (*k*), $k \in$ *Delta*.

În respingerea rezolutivă vom avea:

Termination (*Delta*) = $\square \in$ *Delta*

Rândul doi din procedură cere alegerea din mulţimea de clauze (Delta) a două clauze rezolubile φ (Phi) şi ψ (Psi).

Rândul trei pretinde calcularea rezolventului din clauzele φ şi ψ şi denumirea lui prin χ, eventual prin $\chi_1, \chi_2, \ldots \chi_n$, dacă sunt mai mulţi.

Rândul patru cere să adăugăm la mulţimea Δ de clauze rezolventul nou obţinut, ca urmare a aplicării instrucţiunii din rândul trei.

Alegerea de perechi de clauze, calcularea de rezolvenţi şi adăugarea acestora la setul iniţial de clauze continuă până când este

satisfăcută condiţia de terminare a lui Δ, respectiv până atingem clauza scop, în cazul derivării rezolutive, sau până obţinem clauza vidă, în cazul respingerii rezolutive.

Demonstraţiile, chiar şi cele automatizate, sunt demersuri finite, cu început şi sfârşit. Putem, totodată, să le numim conduite teleologice; ele se încheie când clauza scop a fost atinsă. Căci şi este o clauză scop pentru o derivare dintr-un set de clauze alcătuit din clauzele provenite din premise şi din clauzele provenite din negaţia concluziei.

Instrucţiunile privind alegerea de clauze din Δ şi calcularea rezolvenţilor din clauzele alese sunt îndeajuns de generale şi imprecise, căci nu sunt precizate *criteriile* după care putem face alegerea unei perechi de clauze şi nici *modul* cum „aleg", în fapt „calculez", rezolventul. „Alegerea" rezolventului presupune, de exemplu, *utilizarea unui algoritm* de unificare.

Strategiile rezolutive vor fi cele ce vor preciza ulterior modul cum alegem perechile de clauze din Δ şi modul *cum* calculăm clauzele factor, dacă acesta este necesar sau modul *cum* aflăm efectiv rezolvenţii.

Procedura schiţată mai sus este una generală, în care nu se impun criterii de alegere a perechilor de clauze şi nici algoritmii de unificare.

Procedura dată nu precizează nici cum a fost obţinută Δ, i. e. nu precizează condiţia de început şi nu precizează nici condiţia de terminare. (Noi am detaliat mai sus cele două alternative ale condiţiei de terminare).

Cum putem utiliza metoda rezoluţiei pentru prelucrarea unei baze de cunoştinţe dintr-un domeniu oarecare sau pentru demonstrarea automată a unor concluzii dintr-un set de premise dat? Care sunt etapele strict necesare?

Identificăm, mai întâi, principalele dependenţe şi condiţionări legice sau funcţionale şi le descriem în limbajul logicii predicatelor. Formalizarea ne apare astfel ca o condiţie preliminară a computerizării. În cazul unui raţionament deductiv, prima cerinţă constă în formalizarea premiselor.

Pasul secund constă în aducerea formulelor obţinute din premise la o formă clauzală conform algoritmului specificat în paragraful 7.3.1.

Al treilea pas constă în transpunerea clauzală a concluziei, dacă vrem să aplicăm derivarea rezolutivă, sau în transpunerea clauzală a negaţiei concluziei, dacă vrem să utilizăm respingere rezolutivă.

Cel de-al patrulea pas constă în aplicarea procedurii rezolutive specificate mai sus.

Se înțelege, vom utiliza adecvat condițiile proprii de terminare pentru derivare rezolutivă și cele proprii pentru respingerea rezolutivă, după cum am definit setul inițial de clauze Δ.

Dacă Δ = mulțimea clauzelor provenite din premise, atunci *Termination* (Delta) va fi *Goal* (*k*) și *k* ∈ Delta.

Dimpotrivă, dacă Δ = mulțimea clauzelor provenite din premise și din negația concluziei, atunci *Termination* (Delta) = ∈ Delta.

Succesul unei derivări rezolutive constă în atingerea clauzei scop, iar succesul unei respingeri rezolutive constă în obținerea în Δ a clauzei vide.

j. Univers Herbrand, bază Herbrand și interpretare Herbrand

Suntem îndreptățiți să ne întrebăm ce putere are metoda rezoluției în raport cu o bază de cunoștințe. Presupunând că o clauză *k* este consecință logică dintr-un set de clauze *S* ce alcătuiește baza de cunoștințe, i.e. $S \vdash k$, respectiv că orice model ce satisface *S* satisface și *k*, ne întrebăm dacă există întotdeauna o derivare rezolutivă D = k_1, ..., k_n care să se încheie cu clauza scop *k*.

Altfel pusă problema ne întrebăm dacă, în toate cazurile când dintr-un set de ipoteze *S* se deduce prin scheme de inferență clasice (*modus ponens, modus tollens*, principiul tranzitivității etc.) o concluzie *k*, putem deriva aceiași concluzie prin metoda rezoluției. Este metoda rezoluției completă?

Sau, într-o formulare echivalentă, ne întrebăm dacă întotdeauna când baza de cunoștințe *S* și negația tezei de demonstrat sunt împreună irealizabile putem deriva, din setul de clauze asociat lor, clauza vidă?

Ne întrebăm, deci, dacă ori de câte ori avem un set de clauze irealizabil *S'* (provenită din „premise" și din „negația concluziei"), putem deriva din el cauza vidă . Răspunsul pozitiv la această întrebare îl reprezintă teorema completitudinii deducției rezolutive.

Teorema completitudinii. *Dacă un set de clauze S' este irealizabil, atunci din S' derivă rezolutiv clauza vidă*, i.e. $S' \vdash_{res}$

Cum irealizabilitatea lui *S'* este tot una cu afirmația că *S'* nu are nici un model sau că *S'* este o contradicție, teorema completitudinii poate fi scrisă prescurtat ca:

$Contr(S') \Leftrightarrow S \vdash_{res}$

Demonstrarea acestei teoreme presupune, între altele, utilizarea teoremei lui Herbrand, care afirmă că un set de clauze este irealizabil, dacă și numai dacă există o mulțime finită de clauze de bază Herbrand irealizabile. Va trebui, deci, să introducem un

număr de concepte şi teoreme pregătitoare pentru demonstrarea completitudinii metodei rezoluţiei.

Un set de clauze S este realizabil dacă există un model $M = \langle D, Ic, Iv \rangle$, unde D este un domeniu şi Ic este funcţie de interpretare a constantelor logice, respectiv a constantelor individuale, a simbolurilor funcţionale şi a simbolurilor predicative(vezi paragraful 3 din secţiunea a II-a) şi o asignare de valori Iv acordată variabilelor din clauzele lui S care fac fiecare clauză din S adevărată, iar o clauză din S este adevărată dacă cel puţin un literal l din $k \in S$, interpretat sau instanţiat prin funcţia de asignare Iv este adevărat.

În consecinţă, o interpretare sau un model al unui set de clauze se reduce la o secvenţă de literali instanţiaţi din S.

O mulţime S de clauze este irealizabilă, dacă şi numai dacă este falsă în toate interpretările. Dar noi nu putem epuiza mulţimea domeniilor în care poate fi interpretat un set de clauze. Fantezia creatoare a lui Herbrand ne-a scutit de efortul zadarnic de a încerca să epuizăm mulţimea infinită a domeniilor de interpretare, găsind, sau mai exact construind, un domeniu de interpretare prototip, aşa-numitul univers Herbrand , ce are proprietatea după care un set de clauze S este irealizabil într-un domeniu oarecare, dacă şi numai dacă este irealizabil în universul Herbrand.

D 1. Fiind dat un set de clauze S, vom numi *univers Herbrand* al acelui set o mulţime de obiecte abstracte H(S) construită inductiv din constante individuale şi funcţii de aceste constante, după cum urmează:

1a. H_0 este mulţimea constantelor individuale din S;

1b. Dacă în S nu apare nici o constantă, atunci vom considera pe a drept unica constantă din H_0.

2. Pentru orice $i \geq 0$, fiind dat H_i (numit şi univers de rang i), definim H_{i+1} ca reuniunea elementelor din H_i cu mulţimea elementelor formate din H_i cu ajutorul semnelor funcţionale din S aplicate asupra elementelor din H_i. Astfel dacă f este un semn funcţional ce apare în H_i, având $\delta(f) = n$ şi dacă $t_1, \ldots, t_n$ sunt elemente din H_i, atunci $f(t_1, t_n)$ va fi un element din H_{i+1}.

Exemplul 1.

$S = \{Q(b), \neg P(x) \vee P(f(x))\}$

Atunci:

$H_0 = \{b\}$

$H_1 = \{b, f(b)\}$

$H_2 = \{b, f(b), f(f(b))\}$

$H_\infty = \{b, f(b), f(f(b)), f(f(f(b)))\ldots\}$

Exemplul 2.

$S = \{B(a), O(z) \vee C(z), -I(t) \vee R(c,t)\}$

Atunci:

$H_0 = \{a, c\}$

$H(S) = \{a, c\}$

În exemplul 2 nu avem nici un semn funcțional și, în consecință, expansiunea universului prin inducție nu este posibilă. Dimpotrivă, în exemplul 1 avem un singur element în H_0, dar acesta generează, prin f în H_1, H_2..., alte elemente, prin regulile inducției.

D2. Fie S un set de clauze. Numim *mulțimi de atomi sau bază Herbrand* a lui S mulțimea formulelor obținută din atomii lui S prin substituirea variabilelor din aceștia prin elemente din universul Herbrand al lui S.

Pentru determinarea bazei Herbrand a unei mulțimi de clauze S, construim universul Herbrand al lui S, H(S), identificăm simbolurile predicative din S și în ordinea crescătoare a arității lor după creșterea lui δ, le înlocuim variabilele cu elemente din H(S). Vom nota atomi Herbrand din S sau baza Herbrand din S prin BH(S).

Exemplul 1. $S = \{P(x) \vee Q(x), R(f(a))\}$

$H(S) = \{a, f(a), f(f(a)), \ldots\}$

Simbolurile predicative sunt P, Q, R, toate monadice, i.e.

$\delta(P) = \delta(Q) = \delta(R) = 1$

Baza Herbrand din S, atomii sau instanțierile Herbrand vor fi:

$BH(S) = \{P(a), Q(a), R(a), P(f(a)), Q(f(a)), R(f(a)), P(f(f(a)))\ldots\}$

D3. Numim *instanță de bază* a unei clauze k dintr-un set S o clauză în care variabilele individuale au fost înlocuite cu elemente din universul Herbrand. IB(k) va desemna instanțele de bază ale clauzei k.

Exemplul 1. Fie:

$S = \{P(x), Q(f(y)) \vee R(y)\}$

$k_1 = P(x)$

$IB(k_1) = \{P(a), P(f(a)), P(f(f(a)))\ldots\}$

$k_2 = Q(f(y)) \vee R(y))$

$IB(k_2) = \{Q(f(a)) \vee R(a), Q(f(a))) \vee R(f(a))\ldots\}$

Definim acum o *interpretare Herbrand* a unui set de clauze S.

D4. Fie S un set de clauze, H(S) universul Herbrand al lui S și I o interpretare a lui S în H(S). Vom spune că I este o H-interpretare a lui S dacă I satisface condițiile:

1. I aplică toate constantele din S în ele însele;

2. Dacă f este un simbol funcţional de ordinul n, i.e. $\delta(f) = n$, iar $h_1,\ldots, h_n$ sunt elemente ale lui H(S), atunci lui f i se asociază o funcţie ce aplică $(h_1,\ldots,h_n)$ în $f(h_1, \ldots h_n)$ astfel încât $f(h_1,\ldots,h_n) \in$ H(S).

De notat că într-o interpretare Herbrand nu se pun nici un fel de restricţii asupra simbolurilor predicative din S.

Fie $A = A_1, A_2, \ldots, A_n$ mulţimea atomilor din S sau baza Herbrad a lui S. Atunci o H-interpretare poate fi reprezentată printr-o mulţime I = $\{m_1,\ldots, m_n,\ldots\}$, unde I $\subset$ BH(S) şi pentru orice $j = 1, 2,\ldots,m_j$ este sau A_j sau - A_j , după cum lui A_j i se dă valoarea 1 sau 0.

Deci, fiind dat un set de clauze S având universul Herbrand H(S) nu orice interpretare I a atomilor din S în H(S) este o H-interpretare, ci numai cele care aplică semnele de constante individuale în ele însele şi care asociază oricărui semn funcţional f cu $\delta(f) = n$ o funcţie ce duce n-tuple din H(S), tot în obiecte din H(S).

Un set de clauze S poate fi, evident, interpretat şi într-un alt domeniu decât H(S), cum am văzut în capitolul3 de mai sus dedicat semanticii logicii predicatelor sau în §6.1-6.5, exemplele 2 şi 3 din lucrarea noastră *Logica predicatelor*.

O interpretare Herbrand pentru un set S de clauze se obţine uşor prin valorizarea elementelor din baza Herbrand sau din mulţimea de atomi (Herbrand) asociată lui S. Atomii în apariţie pozitivă descriu cazuri în care atomul ia valoarea 1, iar cei în apariţie negativă descriu cazuri în care acesta ia valoarea 0.

Exemplul 1.

$S = \{P(x) \vee Q(x), R(f(y))\}$

H(S) = $\{a, f(a), f(f(a)),\ldots\}$

BH(S) = $\{P(a), Q(a), R(a), P(f(a)), Q(f(a)), R(f(a))\}$

Vor fi H-interpretări ale lui S interpretările:

$I_1 = \{P(a), Q(a), R(a), P(f(a)), Q(f(a)), R(f(a)), \ldots\}$

$I_2 = \{P(a), - Q(a), R(a), - P(f(a)), Q(f(a)), R(f(a)), \ldots\}$

$I_3 = \{- P(a), - Q(a), - R(a), - P(f(a)), - Q(f(a)), - R(f(a)), \ldots\}$

Întrucât pot exista modele sau interpretări „comune" ale unui set de clauze S într-un domeniu oarecare şi interpretări Herbrand ale aceluiaşi set de clauze se pune problema raportului ce există între aceste tipuri de modele sau interpretări.

Aşa, de exemplu, fiind dată o interpretare M a lui S într-un domeniu D, $M = \langle D, Ic, Iv\rangle$, să se construiască H-interpretarea ce

corespunde interpretării lui S în domeniul D. Aceasta se notează prin M^* (sau I*).

Fie $S_1 = \{- T(x, y) \vee P(x), - M(z, u) \vee P(z), - P(b)\}$

Construim un model $M = \langle D, Ic, Iv \rangle$ unde:

D = {1, 2, 3} este o mulţime de trei elemente:

1. Funcţia Ic asociază lui b elementul 2 din D, i.e. $b = 2$

2. Funcţia Ic asociază predicatelor P, M, T mulţimi sau relaţii definite pe D corespunzător valori de adevăr după cum urmează:

(i) $P(2)$
f

(ii) $M(2, u)$, pentru $u = 1, 2, 3$
f

(iii) $T(2, y)$, pentru y = 1, 2, 3
f

Funcţia Iv ce atribuie valori variabilelor individuale, înlesneşte instanţierile: $x = b = 2$, z = $b = 2$.

Modelul ce face S1 adevărat este definit prin:

(i) $P(2)$ = f i.e. Elementul P nu este adevărat despre elementul 2.

(ii) $M(q, u)$ = f, i.e. Elementul 2 nu se află în relaţia M cu nici un element din D.

(iii) $T(2, y)$ = f, i.e. Elementul 2 nu se află în relaţia T cu nici un element din D.

Obţinem astfel interpretările: - $P(2)$, - $M(2, u)$, - $T(2, y)$ care fac adevărate cele trei clauze din S1, unde u şi y pot lua valorile 1, 2, 3.

Putem face şi mai intuitiv modelul dacă vom interpreta: $P(x) = x$ este părinte; $M(z, u)$ = z este mama lui u; $T(x, y) = x$ este tatăl lui y.

Modelul pentru S1 devine astfel setul de propoziţii instanţiate sau concrete: „Individul 2 nu este părinte", „Individul 2 nu este mamă" şi „Individul nu este tată"

Construim acum o H-interpretare pentru M pe care o vom nota prin M^*.

Pentru aceasta identificăm mai întâi universul Herbrand al lui S1, H(S1) şi baza Herbrand pentru S1, BH(S1).

H(S) = {b}

BH(S) = {$P(b), T(b, b), M(b, b)$}

ţinând cont de interpretarea dată în modelul M constantelor şi semnelor predicative, calculăm valoarea ce revine atomilor din baza Herbrand a lui S1.

P(b) = P(2) = f ⇒ - P(b)	(1, şi (i))
T(b, b) = T(2, 2) = f ⇒ - T(b, b)	(1, (iii))
M(b, b) = M (2, 2) = f ⇒ - M(b, b)	(1, (ii))

Putem acum scrie H-interpretarea ce corespunde modelului *M* pentru *S*1.

$M^* = \{-P(b), -T(b, b), -M(b, b)\}$

Fiind dată o interpretare *M* într-un domeniu *D* a unui set de clauze *S*, o interpretare Herbrand corespunzătoare lui *M*, notată prin *M** este o H-interpretare în care fiecare element d_i din *D* este asociat cu un element h_i din universul lui Herbrand al lui *S*, i.e. din H(*S*) şi în care dacă atomului standard $P(d_1, \ldots, d_n)$ i se atribuie valoarea de adevăr „adevărat" şi atomului Herbrand $P(h_1, \ldots, h_n)$ i se atribuie, la fel, valoarea adevărat.

Fie *S* un set de clauze, $M = \langle D, Iv, Iv \rangle$, o interpretare în care *S* este adevărată. Atunci *S* va fi adevărată în orice H-interpretare *M** ce corespunde lui *M*. Cu alte cuvinte, H-interpretările ce corespund lui *M*, conservă veridicitatea setului de clauze *S*. Invers,

Teoremă **T1.** *Un set de clauze S este irealizabil, dacă şi numai dacă S este fals în orice* H-*interpretare asociată lui S.*

Implicaţia de la stânga la dreapta a enunţului este subînţeleasă, căci irealizabilitatea unui set de clauze înseamnă falsitatea acestuia în orice domeniu şi în orice interpretare, inclusiv în cazul când domeniul coincide cu universul Herbrand şi când funcţia de interpretare satisface cerinţele unei H-interpretări.

Să ne ocupăm acum de implicaţia de la dreapta la stânga. Să presupunem că **a**) *S* este fals în toate H-interpretările şi **b**) că există totuşi o interpretare non-Herbrand *M* în care *S* este adevărată. Atunci înseamnă că putem construi pentru *M*, o H-interpretare *M**, care conservă în universul Herbrand, adevărul lui *M*. Dar aceasta contravine supoziţiei **a**) făcută mai sus, conform căreia *S* este fals în toate H-interpretările. Prin urmare. dacă *S* este fals în toate H-interpretările va fi în mod necesar irealizabil, indiferent de domeniu şi de modul de definire a funcţiei de interpretare.

Concluzia practică ce se impune din teorema T1, demonstrată mai sus, este că pentru a verifica realizabilitatea sau irealizabilitatea unui set de clauze *S* putem utiliza un singur domeniu de interpretare universul Herbrand şi recurge exclusiv la H-interpretări, adică la „atomi" sau elemente din baza Herbrand i.e. elemente din BH(*S*).

O interpretare în sens generic a unui set de clauze *S* este o mulţime de literali instanţiaţi, i.e. literali având variabilele înlocuite cu simboluri desemnând obiecte din domeniul de interpretare *D*.

O clauză de bază *C'*, provenită dintr-o clauză *C* din *S* este satisfăcută de o interpretare I, dacă cel puţin un literal instanţiat din *C'*

apare şi în I, iar o clauză generică este satisfăcută de o interpretare I dacă *orice* instanţiere a lui C, C'_1,..., C'_n este satisfăcută de interpretarea I. Dimpotrivă, o clauză generică C este infirmată de o interpretare I, dacă cel puţin o instanţiere a ei, C', devine falsă în interpretarea I. În sfârşit, un set de clauze S este irealizabil, dacă şi numai dacă, pentru *orice interpretare* I există cel puţin o instanţiere de bază C' a vreunei clauze C din S astfel că C' devine falsă în interpretarea I.

Este util să observăm că în producerea unei interpretări intervin două mecanisme: asignarea de valori din D pentru variabilele individuale din S prin funcţia de asignare Iv (vezi mai sus cap 3 sau paragraful 6.1 din lucrarea noastră *Logica Predicatelor*, 1992.) şi atribuirea de valori de adevăr dată atomilor din S (vezi paragraful 6.3 din *Log. Pred.*), i.e. scrierea lor fără negaţie sau cu negaţie, după cum le asociem valoarea „adevărat" sau „fals"

Mulţimea interpretărilor, inclusiv interpretările Herbrand, pot fi reprezentate prin arbori semantici. Pe această cale putem reda intuitiv procesul construirii unei interpretări, trecerea de la o interpretare parţială la o interpretare completă, legătura dintre clauze şi interpretări, precum mecanismul căutării unei respingeri rezolutive pentru un set S de clauze incompatibile.

k. Arbori semantici, noduri de infirmare şi arbori închişi

Un arbore semantic pentru un set de clauze S este un arbore descendent cu arce etichetate de atomi Herbrand sau conjuncţii de atomi Herbrand respectând condiţiile:

1. Din fiecare nod N pleacă un număr finit de arce. Dacă Q_1, ..., Q_n sunt etichetele arcelor iniţiate în N, atunci $Q_1 \vee \quad \vee Q_n$ este o tautologie logic-propoziţională;
2. Pentru orice nod N definim *intensiunea* lui N, I(N) constând din mulţimea tuturor conjuncţiilor de atomi Herbrand ce etichetează arcele de deasupra nodului N, inclusiv N;
3. În plus, I(N) trebuie să nu conţină perechi opuşi de literali.

Evident, putem să ne imaginăm arbori semantici etichetaţi cu simboluri sau atomi predicativi oarecare, care nu aparţin unei baze Herbrand.

Un astfel de arbore nu face decât să descrie alternativele în care putem interpreta variabilele sau literalii unei formule logice

sau ale unei mulţimi de formule logice descriind o bază de cunoştinţe oarecare.

Fie F(*p, q, r*) o formulă din logica propoziţiilor în care apar variabilele *p, q,* şi *r*. Putem construi pentru F două variante de arbori semantici care respectă definiţia dată mai sus(vezi fig.j1 şi j2 mai jos)

Un arbore semantic complet al unei formule logic propoziţionale poate fi legitim corelat cu atribuirile de valori posibile pentru variabilele propoziţionale. Sunt 2^n drumuri într-un arbore semantic al unei formule cu n variabile, tot atâtea câte atribuiri de valori avem pentru formula în cauză.

Un arbore semantic complet pentru o formulă din logica predicatelor trebuie raportat la alternativele de valorizare pe care le avem pentru instanţierile atomilor predicativi. În arborele din fig j2 de mai jos avem pentru arcele descendente imediat din rădăcină, $p \vee q \vee -p \,\&\, -q$ tautologie şi pentru arcele descendente din p avem $q \vee r \vee -q \,\&\, -r$ de asemenea, tautologie.

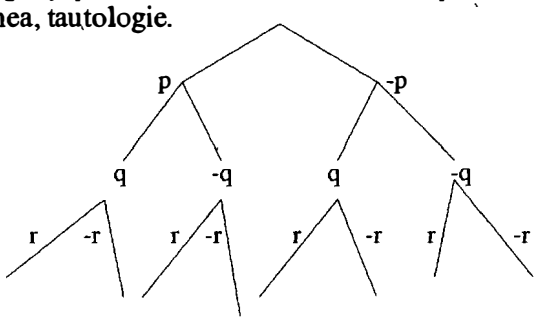


Fig j1 Arbore semantic in logica propoziţiilor(versiunea I-a)

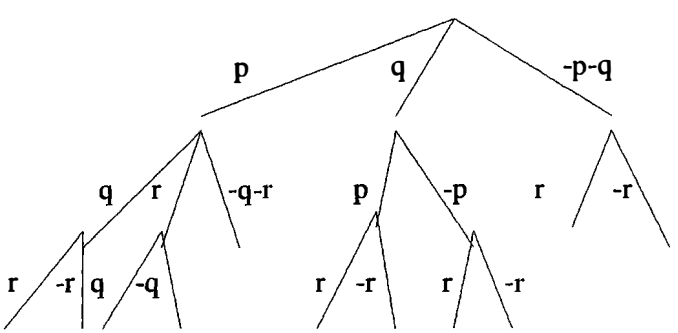


Fig j2 Arbore semantic în logica propoziţiilor (versiunea II-a)

Un arbore semantic este *complet*, dacă fiecare drum către un nod terminal sau „frunză" conține toate elementele din baza Herbrand („atomi") fie în apariție pozitivă, fie în apariție negativă.

Orice drum într-un arbore semantic complet descrie o interpretare completă, iar drumurile incomplete descriu interpretări parțiale.

Exemplul 1.

Fie setul de clauze $S_1 = \{P(x), Q(f(x))\}$

$H(S) = \{a, f(a), f(f(a)) \ldots\}$

$BH(S) = \{P(a), Q(a), P(f(a)), Q(f(a)), P(f(f(a))), Q(f(f(a))) \ldots\}$

Arborele semantic pentru S_1 va fi etichetat cu „atomi" din BH (S), în apariție pozitivă sau negativă.

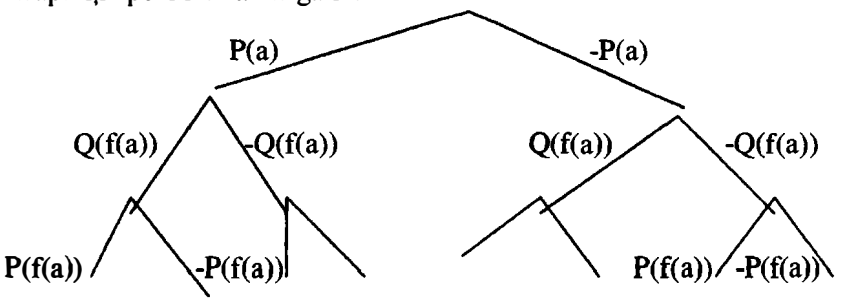


Fig. j3. Arborele semantic al setului S_1 de clauze.

Fiind dat un set de clauze S și o clauză k din S, numim *nod de infirmare* al clauzei k un nod al cărei intensiune sau mulțime de etichete anterioare este identică cu negația clauzei k.

Exemplul 2

1 2 3

Fie $S2 = p \vee q, \neg p \vee q, p \vee \neg q$

și un arbore semantic T_{S2}:

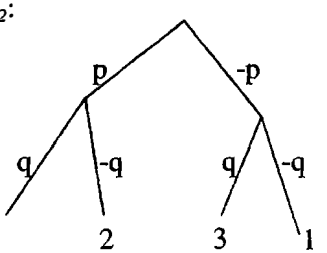


Fig j4. Un arbore semantic pentru formula S_2

Este uşor de observat că nodul 1 din fig. j4 este nod de infirmare pentru clauza 1 din *S2*; la fel nodurile 2 şi 3 din T_{S2} sunt noduri de infirmare pentru clauzele 2 şi 3 din *S2*. Intensiunea nodului 1 este *{-p, -q }* sau $-p \& -q$, ceea ce coincide cu $k_1 = (p \vee q)$, respectiv clauza $k_1 = p \vee q$ devine falsă sau e infirmată pentru atribuirile $p = 0, q = 0$.

Evident, nu e greu să ne imaginăm un nod de infirmare *N* având intensiunea exprimată prin atomi Herbrand sub forma I(*N*) = M ={*P(a), -Q(a), -P(f(a)), Q(f(a))*}, unde fiecare dintre elementele lui *M* etichetează arce situate deasupra nodului *N*, asociat clauzei $k = -P(a) \vee Q(a) \vee P(f(a)) \vee -Q(f(a))$.

Un arbore semantic *T* al unui set de clauze *S* este *arbore închis*, dacă şi numai dacă fiecare nod terminal din *T* este un nod de infirmare al unei claze din *S*.

Exemplul 3

Fie $S3 = k = \{P(a), -P(x) \vee Q(x), -Q(y) \vee R(y), -R(a)\}$ un set de clauze irealizabile. Arborele închis T_s va fi următorul:

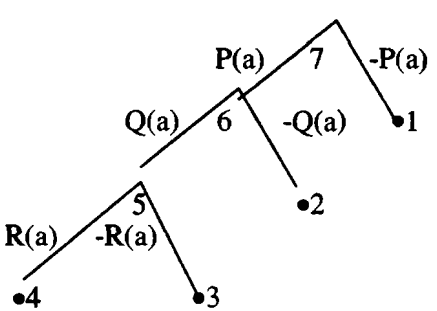


Fig. j5. Arborele de închidere al unui set S de clauze.

Să observăm că toate nodurile terminale N_i $i = 1,\ldots,4$ din arborele T_{S3} sunt noduri de infirmare. Fiecare dintre ele infirmă clauza din *S3* cu acelaşi număr.

De remarcat că arborele închis din fig j5 este un *subarbore* al unui arbore complet alcătuit cu aceiaşi atomi. Arborele complet pentru *S3* presupune dezvoltarea grafului în nodurile 1 şi 2 prin adăugarea ramificaţiilor corespunzătoare atomilor *Q(a)* şi *R(a)* în 1 şi prin adăugarea lui *R(a)* în nodul 2.

Se numeşte *nod de inferenţă* un nod într-un arbore ai cărui succesori imediaţi sunt noduri de infirmare. Nodul 5 în arborele T_S din fig.j5 este nod de inferenţă. Dacă considerăm clauzele 4 şi 5 din S şi facem rezoluţia în literalul R şi $y = a$, atunci nodul 5 va desemna $res_R(k_4, k_3)$.

Identificarea arborelui de închidere la un set de clauze echivalează, aşadar, cu găsirea firului călăuzitor pentru a obţine o respingere rezolutivă pentru setul de clauze S.

1. Teorema lui Herbrand

Cercetările de teoria demonstraţiei întreprinse de Herbrand au stat la baza primelor încercări (Gilmore 1960) de a utiliza ideea realizabilităţii pentru demonstraţia automatizată a unor teoreme. Pornind de la formele normale conjunctive sau de la seturi de clauze acesta a înlocuit variabilele individuale din clauze prin elemente din diferite niveluri ale universului lui Herbrand, construind un program de calculator pentru producerea acestora. Testarea mai departe a realizabilităţii lui S se reduce astfel la decizia unei funcţii logic-propoziţionale.

Ching Liang Chang şi Richard Char Tung Lee în lucrarea lor *Symbolic Logic and Mechanical Theorem Proving* expun teorema lui Herbrand în două versiuni. Prima interpretare face uz de conceptele de arbori semantici şi arbori închişi; cea de a doua se sprijină pe ideile de atomi sau bază Herbrand. Prezentăm mai jos ambele versiuni ale teoremei lui Herbrand. Majoritatea autorilor utilizează cu prioritate cea de a doua versiune (A se vedea în acest sens M.R.Genesereth şi Nils J. Nilsson *Logical Foundation of Artificial Intelligence* 1987, pag. 87).

Teorema lui Herbrand (Versiunea I). *Dacă S este irealizabil, atunci şi numai atunci, orice arbore semantic complet al lui S are un arbore închis.*

Demonstraţie. Fie S un set de clauze irealizabil şi T_S arborele său semantic complet. Fie d un drum complet în T_S iar I_d mulţimea „etichetelor" lui d. Etichetele lui d sunt elemente din baza Herbrand şi definesc o H-interpretare a lui S. Cum S este irealizabil, orice interpretare I_d asociată unui drum trebuie să infirme o clauză de bază c' a unei clauze c din S. Dar cum c' este falsă şi are un număr finit de literali, d trebuie să aibă un nod de infirmare. Cum fiecare drum d din T are un nod de infirmare, putem să-i asociem lui T_S un arbore închis T'_S. Deci, dacă S este irealizabil, arborelui său semantic T_S îi corespunde un arbore închis T'_S.

Invers, fie T_S un arbore semantic complet al unui set de clauze S căruia i-am asociat un arbore închis T'_S Dacă T'_S este un arbore

închis, atunci orice drum din T'_S conţine un nod de infirmare. Cum însă T'_S este un subarbore al lui T_S, rezultă că orice drum terminal în arborele semantic complet T_S va conţine un nod de infirmare ce descrie infirmarea unei clauze de bază c' a unei clauze c din S. Prin urmare, având orice drum complet asociat cu o clauză de bază falsă, setul de clauze S este irealizabil.

O a doua versiune a teoriei lui Herbrand este mai bine cunoscută.

Teorema lui Herbrand (Versiunea a II-a). *Un set de clauze S este irealizabil, dacă şi numai dacă există un set S' de clauze de bază sau baza Herbrand a lui S care este irealizabil.*

Presupunem că S este irealizabil şi că T_S este un arbore semantic complet pentru S. Atunci, potrivit versiunii I a teoremei lui Herbrand, există un arbore semantic închis T'_S care corespunde lui T_S. Fie atunci S' mulţimea clauzelor de bază (sau baza Herbrand) ce devin false în nodurile de infirmare a lui T'_S S' este finită, deoarece T'_S are un număr finit de noduri de infirmare. Deoarece S' este falsă în orice interpretare a sa, S' este irealizabilă.

Invers, să presupunem că S' , baza Herbrand a lui S sau setul său de clauze de bază este irealizabil. Deoarece orice interpretare M a lui S conţine o interpretare M' a lui S', dacă M' infirmă S', atunci interpretarea M infirmă pe S. Cum S' este fals pentru orice interpretare M', rezultă că S va fi fals pentru orice interpretare M a sa.

m. Lema şi teorema ascensiunii

Am văzut până acum cum problema derivabilităţii logice a fost redusă la cea a irealizabilităţii unui set de clauze. La rândul ei, problema irealizabilităţii unui set de clauze oarecare este redusă prin teorema lui Herbrand (versiunea a II-a) la cea a irealizabilităţii unui set de clauze de bază, i.e. a irealizabilităţii atomilor sau bazei Herbrand asociată setului respectiv de clauze.

Lema şi teorema ascensiunii ne arată că pentru orice inferenţă sau deducţie instanţiată, i.e. redusă la exemple sau cazuri particulare, putem construi o deducţie din clauze neinstanţiate, generale.

Lema ascensiunii (Lifting Lemma)

Dacă k_1 şi k_2 sunt două clauze ce nu au variabile individuale comune, k'_1 şi k'_2 sunt instanţe de bază ale lui k_1 şi k_2, r' este un rezolvent din k'_1 şi k'_2, atunci există un rezolvent r din k_1 şi k_2 astfel că r' este o instanţă de bază a lui r.

Demonstraţie: Dacă r' este un rezolvent din clauzele k'_1 şi k'_2 , atunci trebuie să existe un literal P' în k'_1 şi un literal - P' în k_2 astfel că:

$$r' = (k'_1 \setminus \{P'\}) \cup (k'_2 \setminus \{-P'\})$$

Apoi, deoarece k'_1 şi k'_2 sunt instanţe de bază a lui k_1 şi k_2 trebuie să existe o substituţie θ astfel că $k'_1 = k_1\theta$ şi $k'_2 = k_2\theta$. Admiţând că clauzele k_1 şi k_2 conţin mai multe apariţii ale literalului de rezoluţie P şi corespunzător ale lui $-P$ şi că trebuie să construim clauze factor pentru k_1 şi k_2, fie $\{P_1, ..., P_m\}$ mulţimea literalilor din k_1 pe care θ îi aplică în P' şi $\{-P_1, \quad P_n\}$ mulţimea literalilor din k_2 pe care θ îi aplică în $-P'$

Purcedem acum la definirea rezolventului r *din* k_1 şi k_2, ţinând seama de supoziţiile făcute mai sus pentru definirea lui r' Întrucât, atât în k_1, cât şi în k_2 avem apariţii iterate ale literalului de rezoluţie P şi corespunzător $-P$, vom avea nevoie mai întâi de o cea mai generală substituţie unificatoare σ care să unifice pe $\{P_1, ..., P_m\}$ în P'' şi de o cea mai generală substituţie unificatoare τ care să unifice pe $\{-P_1, ..., -P_n\}$ în $-P''$ demn de menţionat este faptul că P'' şi $-P''$ nu diferă între ele numai prin semnul negaţiei, ci şi prin argumentele lor. Fie $\delta = \sigma \cup \tau$ alăturarea celor două substituţii unificatoare de mai sus.

Prin definiţia şi construcţia celui mai general unificator, P', literalul de rezoluţie în obţinerea lui r', trebuie în mod necesar să fie o instanţă a lui P'' şi similar $-P'$, trebuie să fie o instanţă a lui $-P''$.

Fie acum γ cea mai generală substituţie unificatoare între literalii P'', şi $-P''$, care diferă între ei după cum am remarcat mai sus în privinţa argumentelor. Putem acum defini rezolventul r din k_1 şi k_2:

$$r = (k_1 \delta\gamma \setminus \{P_1, \quad P_m\}\delta\gamma \cup (k_2 \delta\gamma \setminus \{-P_1, \quad -Pn\}\delta\gamma)$$

Utilizând definiţia dată lui r, putem rescrie definiţii pentru r' după cum urmează:

$$r' = (k_1 \theta \setminus \{P_1, ..., P_m\}\theta \cup (k_2 \delta\gamma \setminus \{-P_1, ..., -Pn\}\theta)0$$

Deoarece P' este o instanţă a lui P'' şi θ este mai puţin generală decât $\delta\gamma$, r' trebuie să fie o instanţă a lui r.

Lema ascensiunii a fost demonstrată pentru cazul cel mai general posibil, inclusiv pentru cazul când k_1 şi k_2 conţin apariţii iterate ale literalului de rezoluţie şi se impunea construirea unor clauze factor pentru k_1 şi k_2.

Exemplul 1

$k_1 = \{P(x), \ P(f(y)), \ Q(x, z)\}$
$k_2 = \{-P(u), -P(f(t)), \ R(s, u, t)\}$
$k'_1 = \{P(f(a)), \ Q(f(a), z)\}$
$k'_2 = \{-P(f(a)), \ R(s, f(a), a)\}$
$r' = \{Q(f(a), z), \ R(s, f(a), a)\}$
$r = \{Q(f(t), z), \ R(s, f(t), t)\}$

Substituţia θ ce conduce la instanţe de bază este $\theta = \{x = f(y), y = a, u = f(t), t = a\}$.

Substituţiile ce duc la clauzele factor din k_1 şi k_2 sunt $\sigma = \{x = f(y)\}$ şi $\tau = \{u = f(t)\}$. Îmbinarea lor este $\delta = \sigma \cup \tau = \{x = f(y), u = f(t)\}$.

Clauzele factor sunt:

$k_1\sigma = \{P(f(y)), Q(f(y), z)\}$

$k_2\tau = \{-P(f(t)), R(s, f(t),t)\}$

Cel mai general unificator din $k_1\sigma$ şi din $k_2\tau$ este $\gamma = \{y = t\}$.

$k_1\delta\gamma = \{P(f(t)), Q(f(t), z)\}$

$k_2\delta\gamma = \{-P(f(t)), R(s, f(t),t)\}$

$r = \{Q(f(t), z), R(s, f(t), t)\}$

Este uşor de văzut că r' este o instanţiere a lui r prin substituţia $\{t = a\}$.

În ceea ce priveşte redefinirea lui r' pe baza structurii lui r vom avea:

$r' = \{P(x), P(f(y)), Q(x, z)\}\{x = f(y), y = a, u = f(t), t = a\} \setminus$

$\{P(x), P(f(y))\}\{x = f(y), y = a, u = f(t), t = a\} \cup$

$\{-P(u), -P(f(t)), R(s, u, t)\}\{x = f(y), y = a, u = f(t), t = a\} \setminus$

$\{-P(u), -P(f(t))\}\{x = f(y), y = a, u = f(t), t = a\} =$

$\{P(f(a)), Q(f(a), z)\} \setminus \{P(f(a)) \cup \{-P(f(a)), R(s, f(a), a)\} \setminus \{-P(f(a))\} =$

$\{Q(f(a), z), R(s, f(a), a)\}$

Exemplul 2:

$k_1 = P(x)$ $\quad k'_1 = P(a)$

$k_2 = \{-P(y) \vee Q(y)\}$ $\quad k'_2 = \{-P(a) \vee Q(a)\}$

$r = Q(y)$ $\quad r' = Q(a)$

Şi în acest caz r' este o instanţă a lui r.

Lema ascensiunii ne prezintă o cale de generalizare a oricărui caz de inferenţă rezolutivă. Ea ne învaţă să trecem de la instanţieri şi cazuri concrete la o regulă mai generală de inferenţă imediată.

Lema justifică generalizarea doar pentru un pas rezolutiv şi nu pentru o derivare rezolutivă ce cuprinde o mulţime de paşi.

Teorema ascensiunii (Lifting Theorem). *Dacă* S' este un set de clauze de bază ale lui S şi există o derivare rezolutivă r' din S', atunci există o derivare rezolutivă a lui r din S astfel că r' este o instanţiere a lui r.

Demonstraţia se obţine pe baza lemei ascensiunii prin inducţie asupra lungimii derivării rezolutive.

Sunt date:

1. $S' = S\theta$
2. $S' \vdash r'$
3. *lema ascensiunii*

Trebuie de demonstrat că:

4. $S \vdash r$
5. $r' = r\sigma$

Din adevărul lui 2 rezultă că există o derivare rezolutivă D' = $c'_1, c'_2, c'_3, \ldots, c'_n$ astfel încât $c'_n = r'$, $c'_1, c'_2 \in S'$, *iar* $c'_3 = \text{res}_1(c'_1, c'_2)$. Prin lema ascensiunii rezultă că din c_1 şi c_2 derivă rezolutiv c_3 şi că există o substituţie τ astfel că $c'_3 = c_3\tau$. c_3 este astfel primul rezolvent degajat din S conform lemei ascensiunii servind de iniţializare a inducţiei.

Fie acum un element c'_i din D astfel că pentru orice $j < i$, $c'_j \in S'$ sau dacă $c'_j \notin S'$, atunci există m, $k < j$ astfel că $c'_j = \text{res}_1(c'_m, c'_k)$.

Admitem că a fost aplicată lema ascensiunii pentru orice $j < i$ şi că am putut reconstitui derivarea D = $c_1, c_2, c_3 \ldots, c_{i-1}$ prin aplicarea lemei ascensiunii la toţi rezolvenţii din D. Putem acum aplica lema ascensiunii şi pentru obţinerea rezolventului c_i.

Prin aceasta demonstraţia prin inducţie este încheiată.

Putem acum trece la demonstrarea completitudinii metodei rezoluţiei.

n. Teorema completitudinii rezoluţiei

Este metoda rezoluţiei îndeajuns de puternică încât orice teoremă demonstrabilă în logica predicatelor de ordinul întâi (fără relaţia de identitate) să fie demonstrabilă şi prin metoda rezoluţiei? Răspunsul la această întrebare este afirmativ. Principiul rezoluţiei, întregit prin algoritmi de unificare, este complet ca schemă de inferenţă validă.

După cum am văzut anterior problema consecinţei logice semantice este reductibilă la problema testării validităţii formulei ce leagă prin implicaţie premisele de concluzie şi aceasta este echivalentă cu testarea irealizabilităţii setului de clauze obţinut din premise şi din negaţia concluziei. În consecinţă, problema completitudinii principiului rezoluţiei se reduce la demonstrarea faptului că un set de clauze S este irealizabil, dacă şi numai dacă, clauza vidă este derivabilă din S.

Teorema completitudinii rezoluţiei. *Un set S de clauze este irealizabil, dacă şi numai dacă, clauza vidă este derivabilă din el.*

Demonstraţie.

Fie S un set de clauze irealizabil. Notăm prin BH (S) baza sa Herbrand sau mulţimea atomilor lui S, BH(S) = $\{A_1, A_2, A_3 \ldots\}$ şi prin T_S arborele său semantic complet etichetat prin atomii A_1, A_2 , reprezentat ca în figura de mai jos.

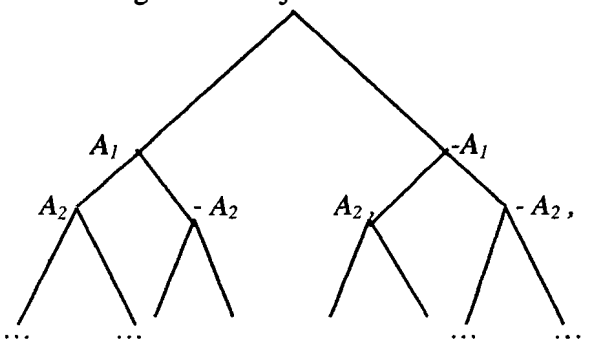


Fig. m1. Reprezentarea unui arbore semantic complet pentru un set de clauze S

Prin teorema lui Herbrand (versiunea I) lui T putem să-i asociem un arbore semantic închis T_S (de fapt acesta este un subarbore al lui T). Dacă T_S conţine un singur nod (rădăcina), atunci clauza vidă trebuie să aparţină lui S, căci nici o altă clauză nu poate fi infirmată în rădăcina unui arbore semantic. În acest caz teorema este, desigur, adevărată.

Să presupunem acum că T_S are mai multe noduri. În acest caz T_S trebuie, în mod necesar, să aibă cel puţin un nod de inferenţă, adică un nod care nu e nod de infirmare, ceea ce ar duce la un drum infinit. Or aceasta contravine supoziţiei după care T_S este un arbore semantic închis şi deci un arbore finit.

Fie N un astfel de nod de inferenţă în T_S şi N_1 şi N_2 nodurile sale imediat descendente sau nodurile sale de infirmare.

Vom nota prin:

$I(N) = \{m_1, m_2, \ldots m_n\}$

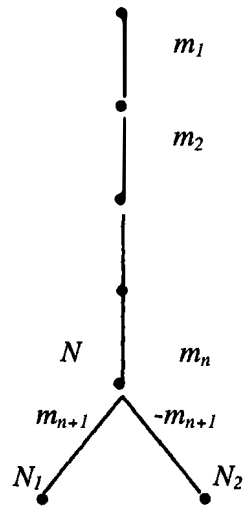
$I(N_1) = \{m_1, m_2, \ldots, m_n, m_{n+1}\}$

$I(N_2) = \{m_1, m_2, \ldots, m_n, -m_{n+1}\}$

intensiunea nodurilor N, N_1 şi N_2 unde m_i, $1 \le i \le n+1$ sunt atomi cu sau fără semnul negaţiei.

Deoarece N_1 şi N_2 sunt noduri de infirmare iar N nu este un nod de infirmare, trebuie să existe două clauze de bază c'_1 şi c'_2

corespunzând la două clauze c_1 şi c_2 din S, astfel că c'_1 este falsă în N_1 şi c'_2 este falsă în N_2, fără ca acestea să fie false în N.



Prin urmare, c'_1 trebuie să conţină pe $-m_{n+1}$ iar c'_2 trebuie să conţină pe m_{n+1} Fie literalul $P'_1 = -m_{n+1}$ şi $P'_2 = m_{n+1}$ Aplicând rezoluţia în literalii P'_1 şi P'_2 respectiv $-m_{n+1}$ şi m_{n+1} obţinem din c'_1 şi c'_2 rezolventul:

$$r' = (c'_1 \setminus \{P'_1\}) \cup (c'_2 \setminus \{P'_2\})$$

Dar r' trebuie să fie fals în I(N) deoarece, atât $c'_1 \setminus \{P'_1\}$, cât şi $c'_2 \setminus \{P'_2\}$ sunt false în I(N). Deoarece r' este un rezolvent din c'_1 şi c'_2, putem obţine, prin lema ascensiunii, din c'_1 şi c'_2 rezolventul r, având ca instanţiere pe r'. Adăugăm apoi rezolventul r la setul iniţial de clauze S şi obţinem setul $S1 = S \cup \{r\}$.

Fie acum T' arborele semantic închis al setului de clauze $S1$, care se obţine din T_S prin eliminarea nodurilor aflate sub nodul unde r' este infirmat. Procesul se repetă din nou obţinând de fiecare dată un arbore de închidere mai scurt cu două arce decât arborele de închidere anterior. Cum numărul arcelor într-un arbore semantic închis este finit, procesul se încheie prin obţinerea unui arbore cu un singur nod (rădăcina) şi, în consecinţă, S trebuie să conţină clauza vidă, căci numai aceasta poate fi infirmată în rădăcina unui arbore semantic. Prin urmare, este derivabilă clauza vidă din S.

Invers, să presupunem că din S este derivabilă clauza vidă. Fie r_1 ...r_k rezolvenţii acestei derivări. Să presupunem că S este realizabil.

Atunci trebuie să existe un model M al lui S. Totuşi, potrivit teoremei rezolventului, dacă un model satisface clauzele c_1 şi c_2, el va trebui totuşi să satisfacă şi rezolventul acesteia. Cum însă clauza vidă este o consecinţă rezolutivă din S, rezultă că S este irealizabil.

Observaţii

În versiunea prezentată mai sus, teorema completitudinii principiului rezoluţiei ca schemă de inferenţă se sprijină, în primul rând, pe teorema lui Herbrand (versiunea I) şi, în al doilea rând, pe lema ascensiunii (Lifting Theorem).

Utilizarea arborilor semantici şi arborilor semantici închişi este de natură să facă demersul raţional mai intuitiv, şi, totodată, să furnizeze un fir călăuzitor în aplicarea principiului rezoluţiei pentru un set irealizabil de clauze oarecare.

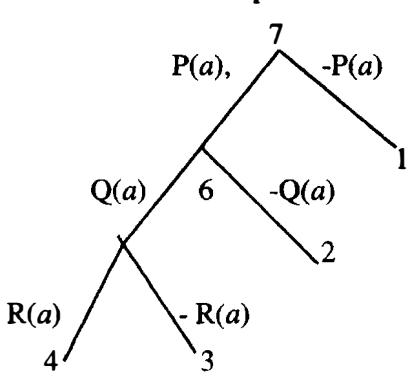
Dacă am construit arborele semantic închis asociat unui set irealizabil de clauze şi am fixat în extremităţile acestuia ("în frunze") clauzele infirmate, atunci putem lesne declanşa un demers rezolutiv pentru obţinerea clauzei vide şi implicit pentru demonstrarea teoremei în cauză.

Un exemplu va fi edificator, pentru a pune în lumină funcţia metodologică a teoremei completitudinii principiului rezoluţiei.

1 2 3 4

Fie $S = P(a), -P(a) \vee Q(x), -Q(y) \vee R(y), -R(a)$

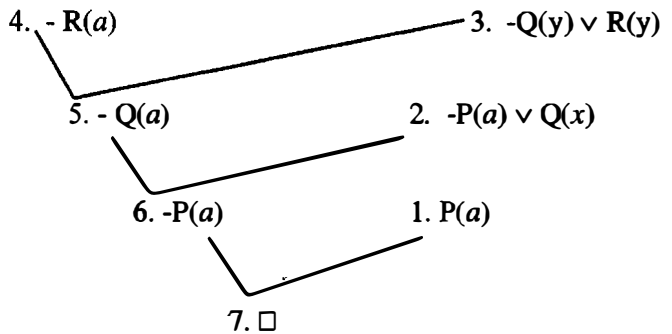
Arborele semantic închis pentru S este arborele din figură:



e)

Fiecare frunză din arbore este un nod de infirmare a unei clauze din S. Nodul 5 este un nod de inferenţă pentru că descendenţii săi imediaţi sunt noduri de infirmare. Fiecare drum în arbore este o

interpretare parţială, iar drumurile 7, 6, 5, 4, şi 7, 6, 5, 3 sunt interpretări semantice complete. Demersul rezolutiv începe cu clauzele asociate nodurilor descendente din nodul de inferenţă, i.e. cu clauzele 4 şi 3 şi continuă ascendent, utilizând rezolventul nou obţinut şi celelalte noduri de inferenţă până atinge în rădăcină clauza vidă.



Este uşor de observat că nodurile interioare ale arborelui semantic închis e) vor fi, pe rând, noduri de inferenţă.

Teorema completitudinii principiului rezoluţiei este, între altele, un instrument util în găsirea unei respingeri rezolutive pentru orice set irealizabil de clauze.

5. Strategii rezolutive în logica predicatelor

Am tratat în paragraful 16.6 strategiile rezolutive la nivelul logicii propoziţiilor. Acolo am prezentat strategia epuizării inferenţelor imediate (the breadth first strategy), strategia mulţimii de suport, rezoluţia semantică, strategia epuizării setului iniţial de clauze, a priorităţii clauzelor unitare, rezoluţia teleologică etc. Toate aceste strategii sunt valabile şi la nivelul logicii predicatelor.

Strategiile sunt nuanţări sau specializări ale metodei rezoluţiei menite să-i sporească eficacitatea. Aplicarea iterată a principiului rezoluţiei este proprie oricărei strategii rezolutive. Strategiile diferă între ele prin condiţiile de start şi condiţiile de terminare, prin criteriile de alegere a perechilor de clauze în care se face rezoluţia, prin modul de divizare sau partiţionare a clauzelor iniţiale, prin criteriile de alegere a literalilor în care se face rezoluţia, prin algoritmii de unificare utilizaţi şi restricţiile ce se pun asupra substituirii variabilelor prin termeni.

Cele mai multe strategii sunt tehnici de respingere rezolutivă eficace. Într-o respingere rezolutivă, setul iniţial de clauze incompatibile *S,'* desemnând datele sau baza de cunoştinţe, provine din două mulţimi de formule: din mulţimea formulelor cu rol de premise sau date explicite iniţiale *S* şi din mulţimea de clauze rezultată din negarea concluziei, *G*, $S' = S \cup G$. Condiţia de terminare într-o respingere rezolutivă este obţinerea clauzei vide din *S'*, $S' \vdash_{res}$

Dimpotrivă, într-o derivare rezolutivă pornim de la clauzele provenite din premise *S* (nu de la *S'*) şi trebuie să obţinem clauza ce reprezintă teza de demonstrat sau clauza scop, *k*, GOAL (*k*) & $S \vdash_{res} k$.

Toate respingerile rezolutive au aceiaşi condiţie de terminare: producerea clauzei vide. Dimpotrivă, derivările rezolutive au drept condiţie de terminare atingerea clauzei scop, care, evident, variază de la o demonstraţie la alta.

Este important de menţionat faptul că alegerea unei strategii rezolutive trebuie să se întemeieze pe analiza şi diagnosticarea naturii problemei, i. e. pe *analiza bazei de cunoştinţe şi a tezei de demonstrat.*

Ce trebuie să ştim despre teza de demonstrat? Mai întâi, dacă e formată dintr-o singură clauză sau e formată dintr-o conjuncţie de clauze. În acest din urmă caz, vom avea de derivat, din baza de cunoştinţe *S,* fiecare dintre clauzele componente a conjuncţiei, iar după aceea va trebui să le asamblăm prin regula introducerii conjuncţiei, 1&.

În al doilea rând, despre baza de cunoştinţe *S* este bine să ştim dacă este alcătuită numai din clauze Horn sau şi din alt tip de clauze.

În al treilea rând, sunt utile întocmirea de tabele cu literalii din *S,* L(*S*) sau cu literali pozitivi din *S,* LP(*S*) şi cu literali negativi din *S,* LN(*S*), precum şi alcătuirea listelor de perechi opuşi de literali, POL (*S*) sau alcătuirea unei diagrame de conexiuni a clauzelor marcând rezoluţiile posibile între acestea şi substituţiile ce le înlesnesc. Pentru teoria demonstraţiei rezolutive sunt, de asemenea, relevante cardinalul lui *S*, precum şi mulţimea literalilor *S'* şi mulţimea clauzelor din *S'*

Tipologia clauzei scop *k*, şi tipologia bazei de cunoştinţe *S* condiţionează strategiile de prelucrare rezolutivă.

Sunt câteva recomandări generale pentru tratarea bazelor de cunoştinţe.

Mai întâi, trebuie *eliminate* din bazele de cunoştinţe *clauzele tautologice.* Într-un caz de respingere rezolutivă o formulă α.(A $\vee$ -A) este infirmabilă, dacă şi numai dacă α este infirmabilă. La fel α.(A $\vee$ -A) este adevărată, dacă şi numai dacă α este adevărată. În consecinţă, A $\vee$ -A nu influenţează decizia formulei α.

La fel, dacă avem două clauze k_1 şi k_2 astfel că prima se subsumează celei de a doua, i. e. $k_1 \subseteq k_2$, putem elimina pe k_1 fără să se schimbe realizabilitatea setului de clauze obţinut, în raport cu realizabilitatea setului iniţial. O clauză k_1 *se subsumează* unei clauze k_2 dacă şi numai dacă există o substituţie s, astfel că $k_1 s \subseteq k_2$, de exemplu $k_1 = \{P(x), Q(y)\}$, $k_2 = \{P(a), Q(y), R(w)\}$ şi $s = \{x/a, y/v\}$.

În sfârşit, în respingerile rezolutive pot fi eliminate din S clauzele care conţin literali puri pentru că din acestea nu pot fi sub nici o formă, reduşi literalii puri. Dacă din S facem derivări rezolutive, nu este oportun să se elimine literalii puri.

Există o mulţime de strategii particulare, ce ne apar că nuanţări ale metodei rezoluţiei. Dintre acestea vom analiza strategia rezoluţiei lini re şi rezoluţia semantică. Pentru nai multe alte strategii cititorul interesat poate recurge la cursul nostru *Logica simbolică şi bazele de cunoştinţe* publicat la Universitatea „Politehnica" în două ediţii în anii 1998 şi 1999.

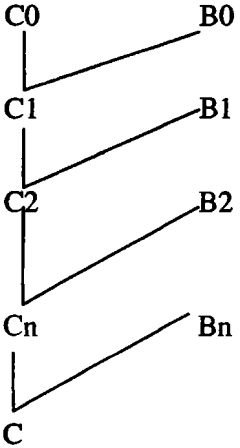
5.1. Rezoluţia liniară

Fie S un set de clauze şi C clauza ce se vrea demonstrată. Numim *rezoluţie liniară* a lui C din S o secvenţă de perechi D = [<C0, B0>, . . . , <Cn, Bn >] care satisface condiţiile:

1. C = Cn+1;
2. C0 şi Bi, cu i ≤ n, sunt elemente din S sau vreun Cj cu j < i;
3. Orice Ci+1, este un rezolvent din Ci şi Bi.

Derivare liniară. Spunem că *C' este derivabil liniar din S* şi notăm $S \vdash_L C$, dacă există o deducere liniară a lui C din S. Spunem că există o *respingere liniară* a lui C din S, dacă din S se derivă liniar clauza vidă, respectiv $S \vdash_L$ Vom nota prin Lin(S) *mulţimea tuturor clauzelor derivabile liniar din S.* Aceast este tot una cu mulţimea tuturor consecinţelor logice deductibile din S prin rezoluţie liniară.

Convenţia de reprezentare grafică a rezoluţiei liniare este una de tip TopDown:



Exemplu de respingere rezolutivă liniară: S = [[p, q], [p, -q], [-p,q], [-p,-q]]. Din $S \vdash_L$

Câteva precizări terminologice. In contextul teoriei rezoluţiei liniare, elementele lui S se numesc *clauze de intrare* (input clauses) sau date iniţiale, premise. Clauzele Ci sau rezolvenţii se numesc *clauze centrale* (center clauses), iar clauzele Bi se numesc *clauze laterale* (side clauses).Este evident că clauzele Ci cunt o submulţime din clauzele de intrare S. Clauzele Ci sunt clauzele din S utilizate pentru obţinerea tezei de demonstrat sau a clauzei scop. Clauza C0 se numeşte clauză de start (starting clause). Ea este doar un element din prima pereche în care se face rezoluţia.

Se acceptă în lumea specialiştilor în teoria demonstraţiei automatizate vorbirea figurativ-antropomorfică despre clauze. De aceea, vorbim într-o derivare rezolutivă de clauze părinţi şi de rezolventul lor direct (copilul lor nou născut) şi de clauze strămoşi şi de rezolvenţii lor mai îndepărtaţi. Clauza de start sau o clauză rezolvent Ci, care sunt clauze centrale interacţionează cu o clauză laterală sau cu un strămoş de al acesteia pentru a genera clauza rezolvent imediat Ci+1. Particularitatea rezoluţiei liniare este faptul că orice rezolvent nou obţinut participă imediat, împreună cu o clauză laterală, la generarea unui nou rezolvent. Rezoluţia liniară reclamă utilizarea ultimei consecinţe obţinute în actul derivativ imediat. Ea este o „străpungere" în adâncime, căci ea trage din concluzii alte noi concluzii.

Orice rezoluție liniară este o rezoluție. Rezoluția liniară este o rafinare a rezoluției. Cum rezoluția este o inferență validă și rezoluția liniară este o inferență validă. Rezoluția liniară este și completă (vezi Anil Nerode, cap II, sect 14).

Să considerăm un raționament din limba naturală, pe care să-l formalizăm în limbajul logicii predicatelor și să-l decidem apoi prin metoda rezoluție utilizând strategia rezoluției liniare.

1. Toți deputații sunt legiuitori.
2. Unii deputați săvârșesc infracțiuni.
3. Cei ce săvârșesc infracțiuni nu respectă legea

---

4. Prin urmare, unii legiuitori nu respectă legea.

Notații

D(x) = „x este un deputat";
L(x) = „x este un legiuitor";
S(y, z) = „y săvârșește z";
I(z) = „z este o infracțiune";
R(u, t) = „u respectă t";
Ll(t) = „t este o lege";

Formalizare

1. $\forall x(D(x) \supset L(x))$
2. $\exists y \exists z(D(y) \wedge S(y, z) \wedge I(z))$
3. $\forall u[\exists t(S(u, t) \wedge I(t)) \supset \exists s(\neg R(u, s) \wedge Ll(s))]$

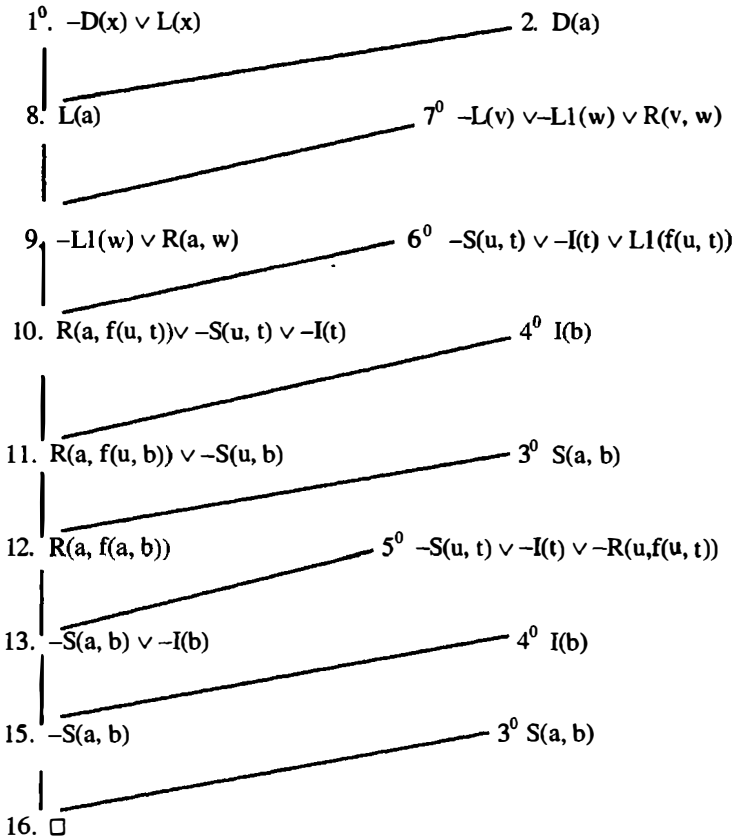
---

4. $\exists w \exists v(L(v) \wedge Ll(w) \wedge \neg R(v, w))$

Clauze

1^0. $\neg D(x) \vee L(x)$ (din 1)
2^0 $D(a)$
3^0 $S(a, b)$ (din 2)
4^0 $I(b)$
5^0. $\neg S(u, t) \vee \neg I(t) \vee \neg R(u, f(u, t))$
6^0 $\neg S(u, t) \vee \neg I(t) \vee Ll(f(u, t))$ (din 3)
7^0. $\neg L(v) \vee \neg Ll(w) \vee R(v, w)$ (din negația lui 4)

Respingere rezolutivă liniară


1⁰. –D(x) ∨ L(x)
2. D(a)
8. L(a)
7⁰ –L(v) ∨–Ll(w) ∨ R(v, w)
9. –Ll(w) ∨ R(a, w)
6⁰ –S(u, t) ∨ –I(t) ∨ Ll(f(u, t))
10. R(a, f(u, t))∨ –S(u, t) ∨ –I(t)
4⁰ I(b)
11. R(a, f(u, b)) ∨ –S(u, b)
3⁰ S(a, b)
12. R(a, f(a, b))
5⁰ –S(u, t) ∨ –I(t) ∨ –R(u,f(u, t))
13. –S(a, b) ∨ –I(b)
4⁰ I(b)
15. –S(a, b)
3⁰ S(a, b)
16. □

Dăm încă un exemplu.

Drama studentului căsătorit

1. Dacă Ştefan învaţă, îşi neglijează soţia.
2. Soţiile neglijate sunt nefericite sau infidele.
3. Dacă Ştefan nu învaţă, rămâne repetent.
4. Soţia lui Ştefan e fericită.

5. Soţii cu soţii infidele sunt încornoraţi.

---

6. Prin urmare, Ştefan rămâne repetent sau Ştefan e încornorat.

Să se decidă asupra raţionamentului prin derivare rezolutivă, strategia rezoluţiei liniare.

Convenţii de notare

a = Ştefan
L(x) = „x învaţă";
C((x, y) = „x este căsătorit cu y";
I(x) = „x este infidel";
N(x, y) = „x neglijează pe y";
C1(x) = „x este încornorat";
F(x) = „x este fericit";
R(x) = „x este repetent";

Formalizare

1. $L(a) \supset (\exists y(C(a, y) \supset N(a, y))$
2. $\forall z\ [\exists u(C(u, z) \wedge N(u, z)) \supset (\neg F(z) \vee I(z))]$
3. $\neg L(a) \supset R(a)$
4. $\exists y(C(a, y) \wedge F(y))$
5. $\forall v[\ \exists t(C(v, t) \wedge I(t)) \supset C1(v)]$

---

6. $R(a) \vee C1(a)$

Aducem formulele 1–5 la formă clauzală.
Prelucrăm formula 1

1. $L(a) \supset (\exists y(C(a, y) \supset N(a, y))$
1a) $\neg L(a) \vee (\exists y(C(a, y) \supset N(a, y))$
1b) $\exists y(\neg L(a) \vee (C(a, y) \supset N(a, y)))$
1c) $\exists y(\neg L(a) \vee \neg C(a, y) \vee N(a, y))$ legea extinderii fictive a domen.
1d) $\neg L(a) \vee \neg C(a, b) \vee N(a, b)$ unui cuantificator ($E\exists$, 4)

Aducem la formă clauzală formula 2.

2. $\forall z\ [\exists u(C(u, z) \wedge N(u, z)) \supset (\neg F(z) \vee I(z))]$
2a) $\forall z\ [\neg \exists u(C(u, z) \wedge N(u, z)) \vee (\neg F(z) \vee I(z))]$
2b) $\forall z\ [\forall u \neg (C(u, z) \wedge N(u, z)) \vee (\neg F(z) \vee I(z))]$

2c) $\forall z[\forall u(-C(u, z) \vee -N(u, z) \vee -F(z) \vee I(z))]$

2d) $-C(u, z) \vee -N(u, z) \vee -F(z) \vee I(z)$

Aducem la formă clauzală formula 3

3. $-L(a) \supset R(a)$

3a) $L(a) \vee R(a)$

Aducem la formă clauzală formula 4.

Eliminarea cuantificatorului existenţial în formula 4. O facem ţinând seama de caracterul monogam al relaţiei de căsătorie în cultura europeană.

4. $\exists y(C(a, y) \wedge F(y))$

4a) $C(a, b) \wedge F(b)$

4a) $C(a, b)$

4b) $F(b)$

Aducem la formă clauzală formula 5.

5. $\forall v[\exists t(C(v, t) \wedge I(t)) \supset C1(v)]$

5a) $\forall v [-\exists t(C(v, t) \wedge I(t)) \vee C1(v)]$

5b) $\forall v [\forall t-(C(v, t) \wedge I(t)) \vee C1(v)]$

5c) $\forall v \forall t (-C(v, t) \vee -I(t) \vee C1(v))$

5d) $\forall v \forall t(-Cv, t) \vee -I(t) \vee C1(v))$

5e) $-C(v, t) \vee -I(t) \vee C1(v)$

Scriem la un loc setul de clauze calculate din formulele 1–5.

1^0 $-L(a) \vee -C(a, b) \vee N(a, b)$

2^0 $-C(u, z) \vee -N(u, z) \vee -F(z) \vee I(z)$

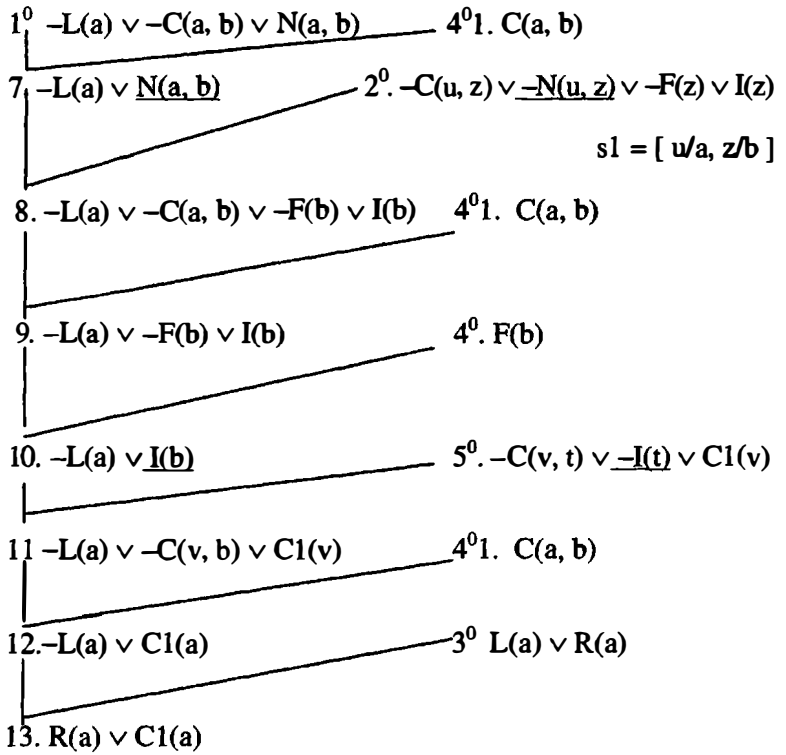
3^0 $L(a) \vee R(a)$

4^01 $C(a, b)$

4^02 $F(b)$

5^0 $-C(v, t) \vee -I(t) \vee C1(v)$

Derivăm rezolutiv din clauzele 1^0 – 5^0 clauza 6 ce descrie concluzia raţionamentului formalizat.


1⁰ –L(a) ∨ –C(a, b) ∨ N(a, b)
4⁰1. C(a, b)
7. –L(a) ∨ N(a, b)
2⁰. –C(u, z) ∨ –N(u, z) ∨ –F(z) ∨ I(z)
s1 = [u/a, z/b]
8. –L(a) ∨ –C(a, b) ∨ –F(b) ∨ I(b)
4⁰1. C(a, b)
9. –L(a) ∨ –F(b) ∨ I(b)
4⁰. F(b)
10. –L(a) ∨ I(b)
5⁰. –C(v, t) ∨ –I(t) ∨ C1(v)
11 –L(a) ∨ –C(v, b) ∨ C1(v)
4⁰1. C(a, b)
12.–L(a) ∨ C1(a)
3⁰ L(a) ∨ R(a)
13. R(a) ∨ C1(a)

Din clauzele obținute din premise am derivat rezolutiv clauza concluzie. Aceasta este o formă de demonstrație orientată înainte sau progresivă. Ea corespunde demonstrației directe din premise a concluziei raționamentului nostru. Propoziția 13 decodificată conduce la concluzia: Ştefan este repetent sau încornorat.

Decideți asupra aceluiași raționament prin respingere rezolutivă și prin metodele Davis–Putnam și a determinărilor semantice succesive.

Testați validitatea raționamentului și în cazul în care vom formaliza prima propoziție astfel:

$L(a) \supset \exists y(C(a, y) \wedge N(a,y))$.

5.2 Strategia rezoluției semantice

Particularitatea rezoluției semantice constă în clasificarea atomilor instanțiați inițiali și a clauzelor ce intră în structura bazei de cunoștințe a unei probleme după modul cum acestea satisfac sau nu un

anumit model. Un model în acest sens este o listă de literali instanţiaţi. Se ia ca model iniţial baza factuală sau atomii instanţiaţi şi modelul se extinde conservativ, apoi, cu toţii literalii instanţiaţi derivaţi rezolutiv.

Pentru decizia prin metoda rezoluţiei semantice urmăm ca şi în alte metode de decizie etapele formalizare, normalizare conjunctivă sau aducere la formă clauzală şi apoi alegem un model, respectiv o listă de atomi instanţiaţi şi apoi împărţim setul de clauze obţinut din premise şi din negaţia concluziei într-o primă submulţime S1 ce satisface setul de clauze şi într-o a doua submulţime S2 ce nu-l satisface.

Numim *rezolvent semantic* un rezolvent într-un literal oarecare care are un părinte în submulţimea S1, care satisface modelul, şi pe celălalt părinte în mulţimea de clauze S2, ce nu satisface modelul M.

Numim *respingere rezolutivă semantică* o derivare rezolutivă a clauzei vide în care toţi rezolvenţii sunt rezolvenţi semantici.

Fiecare rezolvent nou obţinut va fi încadrat imediat ce l-am obţinut în una din cele două clase mutual incompatibile S1 şi S2. Este ceva similar cu declararea sexului unui nou născut.Este „băiat „sau este „fată", respectiv satisface modelul sau nu satisface modelul.

Se înţelege, pentru a putea observa dacă o clauză satisface sau nu un anumit model, va trebui să facem substituţiile unificatoare necesare folosind pentru aceasta, fie intuiţia noastră, fie un algoritm de unificare.

Dăm un exemplu la nivel de set de clauze imcompatibile:

S = [1. $-T(x) \vee -F(x) \vee P(x)$ 2.$-P(y) \vee I(y)$, 3. $-I(z) \vee C(z)$, 4. $-C(a)$,
5. $-L(u) \vee C(u)$, 6. $-P1(u) \vee C(u)$, 7.T(a), 8.F(a), 9.L(a), 10. P1(a)]

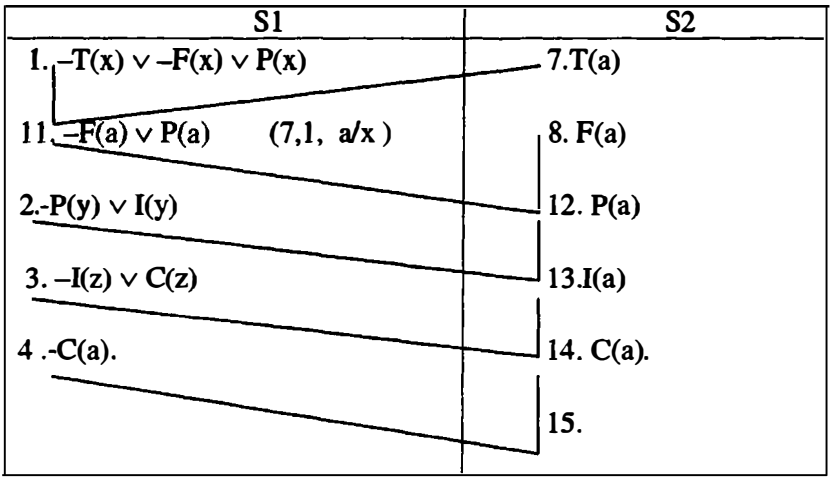
Propunem modelul M = [-C(a), -I(a),-F(a),-P(a),-T(a),-L(a),-P1(a)]
Determinăm mulţimea de clauze S1 ce satisface modelul propus:

S1 = [1. $-T(x)\vee-F(x)\vee P(x)$ 2.$-P(y)\vee I(y)$, 3. $-I(z)\vee C(z)$, 4. $-C(a)$,
5. $-L(u)\vee C(u)$, 6. $-P1(u)\vee C(u)$]

şi mulţimea de clauze S2 ce nu satisface modelul:

S2 = [7.T(a), 8.F(a), 9.L(a), 10. P1(a)]

Construim din S o respingere rezolutivă semantică liniară :



Ilustrăm metoda printr-un alt exemplu.

Unii ageamii sunt înalţi

1. Orice excroc are un ageamiu pe care îl păcăleşte.
2. Toţi samsarii sunt excroci.
3. Unii samsari sunt bucureşteni.
4. Toţi ageamii păcăliţi de samsari sunt înalţi.

---

5. Prin urmare, unii ageamii sunt înalţi.

Să se decidă asupra raţionamentului prin metoda rezoluţiei semantice.

Notaţii

E(x) = „x este excroc";
A(y) = „y este ageamiu";
I(u) = „u este înalt";
S(z) = „z este samsar";
P(x, y) = „x păcăleşte pe y";

Formalizare

1. $\forall x(E(x) \supset \exists y(A(y) \wedge P(x, y))$
2. $\forall z(S(z) \supset E(z))$
3. $\exists u(S(u) \wedge B(u))$
4. $\forall v\ (A(v) \wedge \exists t(S(t) \wedge P(t, v)) \supset I(v))$

---

5. $\exists\ w\ (A(w) \wedge I(w))$

Aducem la formă clauzală fiecare dintre premise:

1. $\forall x\,[E(x) \supset \exists y(A(y) \wedge P(x, y))]$
1a. $\forall x\,[-E(x) \vee \exists y(A(y) \wedge P(x, y))]$ (1, E $\supset$)
1b. $\forall x \exists y\,[-E(x) \vee (A(y) \wedge P(x, y))]$ (EF, 1a)
1c. $\forall x(-E(x) \vee (A(f(x)) \wedge P(x, f(x)))$ (RIIS, 1b)
1d. $-E(x) \vee (A(f(x)) \wedge P(x, f(x))$ (E $\forall$, 1c)
1e. $-E(x) \vee A(f(x))$ 1a LP (E $\wedge$, 1d)
1f. $-E(x) \vee P(x, f(x))$ 1b LP (E $\wedge$, 1d)

unde prin E $\supset$ s–a notat „excluderea implicaţiei"; prin EF regula extinderii fictive a domeniului unui cuantificator, prin RIIS regula a doua de skolemizare şi prin LP legi ale logicii propoziţiilor, în speţă legea distributivităţii disjuncţiei şi legea excluderii conjuncţiei.

2. $\forall z(S(z) \supset E(z))$
2a. $-S(z) \vee E(z)$ (E $\supset$ 2, E$\forall$
3. $\exists u(S(u) \wedge B(u))$
3a $S(a) \wedge B(a)$
3b. $S(a)$ (E&, 3a) 3a
3c. $B(a)$ (E&, 3a)3b
4. $\forall v[(A(v) \wedge \exists t(S(t) \wedge P(t, v))) \supset I(v)]$
4a. $\forall v\,(-A(v) \vee -\exists t(S(t) \wedge P(t, v)) \vee I(v))$ (E$\supset$ + De Morgan 4)
4b. $\forall v\,(-A(v) \vee \forall t\text{-}(S(t) \wedge P(t, v)) \vee I(v))$ (– $\exists$, 4a)
4c. $\forall v \forall t(-A(v) \vee -S(t) \vee \text{-}P(t, v)) \vee I(v))$ (De Morgan +EF, 4b))
4d. $-A(v) \vee -S(t) \vee -P(t, v) \vee I(v)$ 4, (E$\forall$, 4c)

Calculăm clauzele din negaţia concluziei:

5. $-\exists w(A(w) \wedge I(w))$
5a $\forall w\text{–}(A(w) \wedge I(w))$ (– , 5)
5b $\forall w(-A(w) \vee -I(w)$
5c. $-A(w) \vee -I(w)$ 5

Setul de clauze S obţinut din premise şi din negaţia concluziei este:

S	1a $-E(x) \vee A(f(x))$	(1e)
	1b. $-E(x) \vee P(x, f(x))$	(1f)
	2 $-S(z) \vee E(z)$	(2a)
	3a S(a)	(3b)
	3b B(a)	(3c)
	4 $-A(v) \vee -S(t) \vee -P(t, v) \vee I(v)$	(4d)
	5. $-A(w) \vee -I(w)$	(5c)

Căutăm un model pentru setul de clauze S. Acest model va trebui să includă mai întâi literalii instanțiați 3a şi 3b sau propozițiile: „a este samsar „şi „a este bucureştean" Pentru a unifica clauza 3a cu 2 şi cu 4 se impune substituția s= a/z, a/t. Pentru a unifica pe 3a cu 1a şi cu 1b se impune, în plus, substituția a/x. Pentru a unifica pe 1a cu 4 se impune substituția, se impune a/v. Pentru a unifica pe 4 cu 5 se impune substituția a/w. Toate cele 5 variabile individuale din S se vor instanția prin a.

Obținem astfel setul de clauze instanțiate S1.

S1	1a $-E(a) \vee A(f(a))$	(1e)
	1b. $-E(a) \vee P(a, f(a))$	(1f)
	2 $-S(a) \vee E(a)$	(2a)
	3a S(a)	(3b)
	3b B(a)	(3c)
	4 $-A(f(a)) \vee -S(a) \vee -P(a, f(a)) \vee I(f(a))$	(4d)
	5.$-A(f(a)) \vee -I(f(a))$	(5c)

care este un set de cauze instanțiate. Practic, acestea sunt propoziții ca în logica propozițiilor. Prin aceasta am redus o problemă de logica predicatelor, la o problemă de logica propozițiilor. De aici, problema poate fi tratată după exigențele strategiei semantice aplicate la logica propozițiilor expusă în paragraful 16.6.e. din partea consacrată logicii propozițiilor.

Este mai firesc, însă, să începem cu modelul parțial furnizat de clauzele factuale inițiale şi să încercăm să-l extindem treptat.

M1 = [S(a),B(a)]

Includem în clasa S_T aceste prime două clauze şi, aplicând principiul rezoluției, încercăm să extindem modelul. Prin rezoluție

între 3a şi 2 făcând uz de substituţia unificatoare s= a/z conchidem asupra lui E(a). Scriem:

M2 = [S(a),B(a), E(a)]

Modelul M2 acoperă clauzele 2,3a şi 3b. Cum E(a) este adevărat obţinem prin rezoluţie din E(a) şi clauzele 1a şi, 1b printr-o substituţie unificatoare s = a/x, literalii A(f(a)), P(a, f(a), pe care-i adăugăm la modelul anterior M2 şi-i extindem la modelul M3.

M3 = [S(a),B(a), E(a), A(f(a)), P(a, f(a)]

Acesta acoperă în plus clàuzele 1a şi 1b. Mai departe, clauza unitară A(f(a)), proaspăt închisă în model unifică cu clauza 4. Pentru salvgardarea clauzei 4 (sau cuprinderea ei în model) trebuie să includem în model pe I(f(a))

M4 = [S(a),B(a), E(a), A(f(a)), P(a, f(a), I(f(a))]

In acest caz, clauza 6 nu va mai putea fi acoperită de modelul nostru şi de nici un alt model, deoarece prin substituţia unificatoare obţinem doar opuşi ai literalilor din M4.

Desigur, noi putem construi un model parţial care să acoperă clauza 6., dar în acest caz nu vom acoperi cel puţin o altă clauză. „Pătura" (sau modelul) este prea scurtă, când ne acoperim la umeri, rămânem descoperiţi la picioare sau invers.!

Sugestie. Refaceţi exemplul de mai sus luând drept model parţial iniţial pe M = [-S(a), -B(a)]

CONCLUZII

Am prezentat în acest volum două teorii logice, logica propoziţiilor şi logica predicatelor , teorii care alcătuiesc temelia pe care se edifică celelalte sisteme sau de la care plecând prin suspendarea unor principii sau prin adăugarea altora se construiesc alte sisteme logice.

Mai multe metode de decizie sunt prezentate în premieră în limba română. Aşa sunt metoda arborilor de decizie, metoda rezoluţiei şi strategiile rezolutive, metoda rezoluţiei duale. Altele sunt noi, credem, şi pentru cititorul de peste hotare:metoda interpretărilor semantice succesive, interpretarea sistemică a formelor normale

conjunctive, strategia rezoluției teleologice, schemele de inferență și dualitatea metodelor rezolutive.

Tratarea axiomatică a logicii se poate întreprinde cu mai mult folos după ce studentul este acomodat cu sintaxa și semantica teoriilor logice de bază, după ce acesta are cunoștințe și de aplicațiile logicii în viața cotidiană și în construirea bazelor de cunoștințe relaționale.

Tratând analitic mai multe metode de decizie am urmărit acomodarea studentului cu aplicațiile logicii. Logica modernă nu mai poate fi tratată în stilul scolasticii medievale. Ea nu trebuie memorată pe dinafară, ca unele date istorice sau ca unele date din geografie. Schemele ei de inferență ca și unele metode, cum sunt formalizarea, modelarea, axiomatizarea, metodele de decizie etc., devin prin exersare tehnici de gândire care-l apropie pe tânăr de utilizarea logicii pentru rezolvarea unor clase de probleme.

Metateoria, ca o reflecție de rangul doi, presupune cunoașterea limbajelor și teoriilor logice de bază. Ea poate fi mai bine abordată după ce anterior ai prezentat și tratarea axiomatică a acelor teorii.

Ca și înotul, logica se învață prin exersarea tehnicilor și metodelor ei, prin exercițiu cotidian.

Asimilarea logicii este ușoară, dacă te obișnuiești să o faci cu creionul în mână și cauți să înțelegi fiecare concept și fiecare metodă. Studiul ei trebuie întreprins cu continuitate și, din aproape în aproape, nu poate fi amânat până la câteva săptămâni înainte de examen.

Pentru a înlesni o atitudine activă în studierea logicii am anexat câteva liste de subiecte curente de logică. Am preferat să dăm subiecte locale, restrânse și precise cu rezolvări precise scurte, decât probleme complexe, cu formalizări dificile. Dispunem, desigur, și de clase de probleme mai complexe pentru studenții mai avansați în studierea disciplinelor logice.

Volumul următor va trata problemele axiomaticii logicii propozițiilor și logicii predicatelor, metateoria limbajelor logice clasice, demonstrațiile de noncontradicție, completitudine, independență, decidabilitate, marile teoreme despre limitele formalismelor.

Un loc aparte vom rezerva teoriei moderne a conceptului, în special teoriei definițiilor, diviziunilor și clasificărilor. Acestea vor fi regândite dintr-o perspectivă modernă, corelată cu programarea logică, cu metoda rezoluției duale.

Totodată, vom prezenta mai multe teorii logice neclasice, cum sunt logicile modale şi temporale, logicile dinamice, epistemice, deontice, logica acţiunii, teoria logică a scopurilor şi programelor.

Un domeniu care a devenit în ultimul deceniu deosebit de activ şi promiţător este logica inductivă, corelată astăzi cu rezoluţia inversă, cu teoria argumentării şi, evident, cu tehnica construirii teoriilor ştiinţifice pornind de la date empirice.

Tot de acest domeniu ţin logicile fuzzy şi cele probabiliste, teoria raţionamentului prin analogie. Dar despre toate aceste cu alt prilej.

BIBLIOGRAFIE SELECTIVĂ

1. Ben-Ari, M., *Mathematical Logic for Computer Science*, Prentice Hall, London, 1993.
2. Benthem, J.F.A.K. van, *Exploring logical Dynamics*, CSLI Publications, 1996.
3. Beth, Evert, *Formal Methods*, D.Reidl Publishing Company, 1962.
4. Boolos, George S., *Computability and Logic,* Cambridge University Press, 1995.
5. Botezatu, Petre *Constituirea logicităţii*, Editura Ştiinţifică şi Enciclopedică,1983, pp 95-101.
6. Botezatu, Petre *Semiotică şi negaţie*, Editura Junimea, 1974, p37-51.
7. Chang, C.L. , Lee, R.C.T., *Symbolic Logic and Mechanical Theorem Proving*, Academic Press, 1973.
8. Church, Alonzo, *Introduction to Mathematical Logic,* Part 1, Princeton,1944.
9. Didilescu, Ion, Botezatu, Petre, *Silogistica,* Editura Didactică şi Pedagogică, 1976.
10. Ebbinghaus,H.D., Flum,. W, J.Thomas, *Mathematical Logic,* Springer Verlag
11. Enescu, Gh, *Dicţionar de logică*, Editura Ştiinţifică şi Enciclopedică, 1985.
12. Enescu, Gh, *Tratat de Logică*, Editura Lider, 1997.
13. Epstein, Richard L., *Predicate Logic*, Oxford University Press, Oxford, 1994.
14. Gärdenfors, Peter, *Knowledge in Flux, Modeling the Dynamic Epistemic States*,The MIT Press, 1988.
15. Genesereth,M.R.,Nilsson,Nils *Logical Foundation of Artificial Intelligence*, Morgan Kaufmann Pubishing, 1987.

16. Kleene, S. C. *Mathematical Logic, John* Waley & Sons, New York,1967.
17. Loveland, Donald, W., *Automated Theorem Proving, A Logical Basis*, North Holland Publishing Company, 1978.
18. Ligeza, A., *Logical Foundations for Knowledge Based Control Systems. Knowledge, Representation, Reasoning and Theoretical Properties*, Raport LAAS, no 92269, Juillet 1992, Laboratoires d'Automatique et d'Analyse des Systèmes, Toulouse.
19. Maliţa, Mihaela, Maliţa, Mircea, *Bazele inteligenţei artificiale*, 1, Logici propozitionale, Editura Tehnică,1987.
20. Metakides, G., Nerode, A., *Principii de logică şi programare logică*, Editura Tehnică, 1998.
21. Miroiu, Adrian, *Introducere în Logica Filosofică*, vol 1. *Logică şi formalizare*, Editura Universităţii Bucureşti, 1994.
22. Nerode,Anil , Shore , RichardA. , *Logic for Applications*, Springer, 1997.
23. Nilsson,Nils, *Problem Solving in Artificial Intelligence*, Mc Graw Hill, New York , 1971.
24. Plantinga, Alvin, Natura necesităţii, Editura Trei, 1988.
25. Popa, Cornel, *Logica predicatelor*, Editura Hyperion, 1992.
26. Popa, Cornel, *Logica simbolică şi bazele de cunoştinţe*, I, II, Universitatea „Politehnica" Bucureşti, 1998, reeditat 1999.
27. Popa, Cornel , *Teoria definiţiei*, Editura Ştiinţifică, 1972.
28. Popa, Cornel ,*Teoria acţiunii şi logica formală*, Editura Ştiinţifică şi Enciclopedică, 1984.
29. Ramsay, Alan, *Formal Methods in Artificial Inteligence*, Cambridge University Press, 1988.
30. Shan-Hwei Nienhuys-Cheng, de Wolf, Ronald, *Foundations of Inductive Logic Programing*, Springer, 1997.
31. Suppes, Patrick, *Introduction to Logic*, D.van Nostrand Company, Princeton 1957.
32. Tîrnoveanu Mircea, *Elemente de logică matematică*, Editura Didactică şi Pedagogică, 1964.
33. Wang Hao, *Studii de Logică Matematică*, Editura Ştiinţifică, 1972.

ANEXE

A. *SUBIECTE TEORETICE DE LOGICA PROPOZIȚIILOR*

1. Limbajul logicii propozițiilor și limbile naturale. Analiza și sinteza formulelor bine formate
2. Propoziție și semn propozițional
3. Principiile logicii propozițiilor
4. Negația logică și exprimarea ei în limbile naturale
5. Conectivele logice binare și teoria funcțiilor de adevăr
6. Raporturile logicii cu teoria cunoașterii, științele cognitive și informatica
7. Gândire, limbaj, logică. Niveluri și accepții ale termenului „logică"
8. Logica și teoriile științifice
9. Logica clasică și logicile neclasice. Clasificarea sistemelor de logică neclasică
10. Logica și legile gândirii
11. Logica și activitățile umane
12. Formule și funcții logice
13. Tabloul funcțiilor logice bivalente
14. Interdefinibilitatea conectivelor logice. Sisteme complete de funcții logice
15. Noțiunea de consecință logică. Consecința logică sintactică și consecința logică semantică
16. Implicanți și consecințe logice
17. Interpretare și model în logica propozițiilor
18. Definiția logic-propozițională a pătratului logic
19. Pătratul logic și tabloul funcțiilor logice bivalente
20. Clasificări ale funcțiilor logice
21. Teoria mulțimilor și funcțiile logice
22. Metoda matriceală
23. Metoda lui W.O. Quine
24. Metoda tablourilor semantice
25. Metoda arborilor de decizie

26. Formele normale conjuntive şi cauzele disjunctive
27. Formele normale disjunctive şi clauzele conjunctive
28. Formele normale şi conceptele de model şi contramodel
29. Realizabilitate şi validitate
30. Infirmabilitate şi contradicţie
31. Teorema rezolventului. Enunţ şi demonstraţie
32. Schemele clasice de inferenţă şi principiul rezoluţiei
33. Derivare şi respingere rezolutivă
34. Rezoluţia liniară
35. Rezoluţia semantică
36. Clauza unitară şi strategia clauzei unitare
37. Rezoluţia teleologică
38. Strategia mulţimii de imput
39. Metoda interpretărilor semantice succesive
40. Metoda Davis–Putnam
41. Metoda axiomatică. Etapele construirii unui sistem axiomatic
42. Sistemul axiomatic Hilbert Ackermann
43. Demonstraţia de noncontradicţie a sistemului Hilbert-Ackermann
44. Demonstraţia de completitudine a sistemului Hilbert Ackermann
45. Independenţa sistemului Hilbert Ackerman
46. Teorema deducţiei şi demonstraţiile în sistemul Hilbert Ackermann
47. Schemele de inferenţă şi calculul natural
48. Calculul secvenţial în logica propoziţiilor
49. Teoria demonstraţiei. Demonstraţia directă şi reducerea la absurd
50. Demonstraţie şi argumentare
51. Arborii de derivare şi demonstraţia
52. Baze de cunoştinţe,problemă, soluţie
53. Clauzele Horn şi instrucţiunile Prolog
54. Limitele logicii propoziţiilor
55. Semantica logicii propoziţiilor şi reprezentarea grafică a stărilor descrise de propoziţii
56. Algebrele booleene şi logica propoziţiilor. Principalele legi şi teoreme de de algebră booleană
57. Principiul dualităţii şi regulile dualităţii
58. Rezoluţia duală. Specii de rezolutie duală
59. Principiul rezoluţiei duale. Teorema rezolventului
60. Formele normale, bazele de cunoştinţe şi demonstraţia automatizată
61. Principiul dualităţii şi rezoluţiile duale

B. *EXERCIŢII DE LOGICA PROPOZIŢIILOR*

1. Se dă formula: $(p \supset q) \wedge (q \supset r) \rightarrow (p \supset r)$. Să se aducă la forma normală conjunctivă.
2. Se dă formula: $((p \supset q) \wedge (r \supset q)) \supset ((p \vee r) \supset q)$. Să se decidă asupra ei prin arbori de decizie.
3. Se dă formula: $((p \vee q) \rightarrow r) \rightarrow ((p \wedge -r) \rightarrow (q \rightarrow r))$. Să se aducă la forma sa normală disjunctivă.
4. Se dau propoziţiile:

a. Dacă afacerile îmi merg bine, îmi construiesc o vilă.
b. Dacă îmi construiesc o vilă, o voi ridica la Costineşti, la Predeal sau la Snagov.
c. Dacă îmi construiesc o vilă, o voi face din cărămidă sau din prefabricate.
d. Dacă vila e construită din prefabricate, vor emana în interior substanţe toxice.

Să fie formalizate în limbajul logicii propoziţiilor.

5. Se dau propoziţiile:

a. Dar vila mea nu va emana în interior substanţe toxice.
b. Afacerile îmi merg bine.
c. Dacă îmi ridic vila la Costinesti sau la Predeal, va fi departe de casa mea din Bucureşti.
d. Dar vila mea nu va fi departe de casa mea din Bucureşti.

Să se formalizeze în limbajul logicii predicatelor.

6. Se dau formulele:

1'. $p \supset q$
2'. $q \supset (r \vee s \vee t)$
3'. $q \supset (r1 \vee s1)$
4' $s1 \supset v1$
5'. $-v1$
6'. p
6. $(s \vee t) \supset d$
7. $-d$

Să se construiască o interpretare în limba română pentru ele.

7. Se dă setul de clauze disjunctive :

$S3 = (-p \vee q) \wedge (-q \vee r \vee s \vee t) \wedge (-q \vee r1 \vee s1) \wedge (-s1 \vee v1) \wedge -v1 \wedge p \wedge$
$\wedge (-r \vee d) \wedge (-s \vee d) \wedge -d$

Să se construiască un model pentru el, utilizându-se una dintre metodele: Davis-Putnam, metoda interpretărilor semantice succesive, sau metoda rezoluției.

8. Se dă formula $\psi = ((-p \vee q) \wedge (r \supset p)) \vee (p \supset (t \vee -q))$
Să se decidă asupra ei prin metoda lui W. O. Quine.

9. Se dau clauzele:

$$S = \{p \vee s, q \vee -s, -p \vee t, r \vee -t, -r \vee v, -q\}$$

$$k = v$$

Să se derive clauza k = v.

10. Se dă setul de clauze:

$$KB = \{p \vee s, q \vee -s, -p \vee t, r \vee -t, -r \vee v, -q, -v\}$$

Să se respingă rezolutiv setul.

11. Se dau propozițiile:

1. Dacă Ştefan termină facultatea, va lucra în industria electronică sau în cercetare.
2. Dacă va lucra în industria electronică, va controla execuția proceselor tehnologice.
3. Dacă va lucra în cercetare, va face descoperiri sau invenții.
4. Dar Ştefan nu va controla execuția proceselor tehnologice şi nu va inventa sau descoperi ceva.
5. Prin urmare, Ştefan nu va termina facultatea.

Să se formalizeze în limbajul logicii propozițiilor şi să se decidă prin metoda rezoluției asupra validității sale.

12. Se dau formulele:

1. $p \supset (q \vee r)$
2. $q \supset s$
3. $r \supset (t \vee v)$
4. $-s \wedge -(t \vee v)$
5. $-p$

Să se construiască un raționament pe structura lor. Exprimați în limba română acest raționament.

13. Se dă setul de clauze:

$$S = \underset{1}{(p \vee q)} \wedge \underset{2}{(-p \vee r)} \wedge \underset{3}{-r} \wedge \underset{4}{(-q \vee s)} \wedge \underset{5}{(-s \vee r)}$$

Să se decidă asupra lui prin metoda interpretărilor semantice succesive.

14. Se dau propoziţiile:
Dacă oboseşti, te culci sau bei cafea.
Dacă munceşti, oboseşti.

Prin urmare, dacă munceşti şi nu bei cafea, te culci.

Să se formalizeze în limbajul logicii propoziţiilor şi să se decidă asupra validităţii raţionamentului .

15. Se dau clauzele:
$S = (-p \vee q \vee r) \wedge (-s \vee p) \wedge s \wedge p) \wedge s \wedge -r \wedge -q.$
Să de decidă asupra lor prin respingere rezolutivă.

C. EXERCIŢII DE FORMALIZARE. DEFINIŢII DE PREDICATE ŞI PROPOZIŢII DIN PREDICATE PRIMARE DATE

1. Se dau predicatele: I(x, y) = „x iubeşte pe y"; P(x, y) = „x este un părinte al lui y"; B(z) = „z este bărbat"
Să se scrie cu ajutorul acestora, în logica predicatelor, propoziţiile:
a. Există un bunic îndrăgostit;
b. Unii părinţi sunt fericiţi, dacă vreun copil de al lor se îndrăgosteşte.

2. Se dau predicatele: I(x, y) = „x iubeşte pe y"; C(x, y) = „x este căsătorit cu y". Să se formalizeze în logica predicatelor: 1. Nu-i adevărat că toţi îşi iubesc soţiile. 2. Unii iubesc soţiile altora.

3. Se dă predicatul V(x, y, z) = „x vinde lui y, z" Să se definească predicatele: C(x, y, z) = „x cumpără de la y , z" şi C1(y) = „y cumpără câte ceva de la fiecare"

4. Se dă predicatul V(x, y, z) = „x vinde lui y, z" Să se scrie in logica predicatelor propoziţia:
a. Cineva cumpără de la altcineva ceva;
b. să se definească predicatul unar C(y) = „y cumpără de la cineva totul"

5. Se dă predicatul V(x, y, z) = „x vinde lui y, z". Să se scrie predicatul „x vinde tuturor câte ceva" şi să se exprime în limba română înţelesul formulei: $\forall x \exists y \forall z V(x, y, z)$.

6. Se dă predicatul S(x, y, z) = „x spune lui y, z" să se definească noţiunile de vorbitor şi de ascultător şi să se exprime in limba română înţelesul formulei: $\exists x \forall y \exists z$ S(x, y, z).

7. Se dă predicatul S(x, y, z) = „x spune lui y, z" Să se formalizeze în logica predicatelor propoziţia: Petru spune lui Pavel ceva şi lui Nicu altceva, şi să se definească predicatul vorbitor (x) ("spunător").

8. Se dă predicatul S(x, y, z) = „x spune lui y, z". Să se definească noţiunea de mesaj sau ceea ce se spune şi noţiunea de secret.

9. Se dă predicatul S(x, y, z) = „x spune lui y, z" Aflaţi câte propoziţii putem forma prin cuantificare pornind de la predicatul S(x, y, z). În plus, definiţi prin cuantificarea aceluiaşi predicat predicatul ascultător (y).

10. Se dau predicatele S(x, y, z) = „x spune lui y, z" şi fals(x) = „x este fals". Definiţi predicatul: mesaj (z) sau ceea ce se spune şi predicatul minciună(z) = z este o minciună.

11. Se dau predicatele: iubeşte(x, z) şi se *teme*_de(x, y) = x se teme de y. Să se formalizeze propoziţia:
a. Cei iubiţi de toţi nu se tem de nimeni;
şi să se definească predicatul:
b. curajos(x) = „x este un curajos"

12. Se dau predicatele: iubeşte(x, z) şi se teme de (x, y) = x se teme de y. Să se definească cu ajutorul predicatelor date propoziţiile:
a. Cel care iubeşte pe cineva nu se teme de nimeni.
b. Cel care este iubit de toţi nu se teme de nimeni.

13. Se dau predicatele: iubeşte(x, z) şi se *teme* de(x, y) = x se teme de y. Să se definească cu ajutorul predicatelor date propoziţiile:
a. Fricoşii nu sunt iubiţi de nimeni.
b. Îndrăgostiţii sunt curajoşi.

14. Se dau predicatele: păzeşte(x, z) = „x păzeşte pe z" şi se *teme* de(x, z) = x se teme de z. Să se definească cu ajutorul predicatelor date propoziţia:
a. Cel ce este păzit se teme de cineva;
şi predicatul:
b. gardian(x) = „x este un gardian.

15. Se dau predicatele: iubeşte(x, z) şi căsătorit (x, y) = x este căsătorit cu y. Să se definească cu ajutorul predicatelor date predicatele:
a. soţ adulter(x) = „x este un soţ adulter";
b. soţie infidelă(x) = y este o soţie infidelă.

16. Se dau predicatele: iubeşte(x, z) şi căsătorit (x, y) = x este căsătorit cu y. Să se definească cu ajutorul predicatelor date predicatele:
a. soţ iubitor(x) = „x este un soţ iubitor".
b. soţ încornorat(x) = „x este un soţ încornorat „.

17. Se dă predicatul : împrumută(x, y, z) = „x împrumută de la y, z" Să se definească predicatele:
a. creditor(x) = „x este un creditor"
b. împrumut (z) = „z este un imprumut.

18. Se dă predicatul: angajează (x, y). Să se definească predicatele:
a. patron(x) = „x este un patron"
b. salariat(x) = „x este un salariat"

19. Se dau predicatele: vecin(x, y) , căsătorit(x, y), are(x, y), poartă(x, y), roşu(u), femeie(z) , bărbat(t), rochie (v), cățel (w).
Să se scrie propoziţiile:
a. Vecinul lui Ion are un cățel.
b. Vecina lui Petru poată o rochie roşie.

20. Se dau predicatele: vecin(x, y) , căsătorit(x, y), are(x, y), poartă(x, y), roşu(u), femeie(z), bărbat(t), rochie (v), pălărie(u).
Să se scrie propoziţiile:
a. Proprietarul este căsătorit şi poartă pălărie.
b. Vecinul meu este căsătorit cu femeia ce poartă pălărie roşie.

21. Se dau predicatele: vecin(x, y) , căsătorit(x, y), are(x, y), poartă(x, y), roşu(u), femeie(z) , bărbat(t), rochie (v), pălărie(u).

Să se scrie propoziţiile:
a. Cel ce poartă pălărie neagră este căsătorit cu femeia ce poată o rochie roşie.
b. Soţul vecinei mele nu poartă pălărie.

22. Se dau predicatele: angajează (x, y), : vecin(x, y) , căsătorit(x, y), are(x, y), poartă(x, y), roşu(u), femeie(z) , bărbat(t), rochie (v), pălărie(u), căţel (w).
Să se formalizeze propoziţiile:
a. Eu am un vecin patron a cărui soţie are un căţel roşu.
b. Soţia mea nu poartă rochii roşii.

23. Se dau predicatele: ciupercă(x) = „x este o ciupercă"; roşu (u) = „u este roşu"; otrăvitor(y) = „y este otrăvitor"; angajează (x, y), mănâncă(x, y) .
Să se scrie propoziţiile:
a. Toate ciupercile roşii sunt otrăvitoare.
b. Patronii nu mănâncă ciuperci, dar unii dintre salariaţii lor mănâncă ciuperci.

24. Se dau predicatele: naşte(x, y); căsătorit(z, x).
Să se definească predicatele:
a. mamă(x, z)
b. copil(u).

25. Se dau predicatele: naşte(x, y); căsătorit(z, x).
Să se definească predicatele:
a. bunică(x, w)
b. x este soţ şi părinte de copil căsătorit.

26. Se dau predicatele: iubeşte(x, z); om(x); măcelar(z), femeie (w), vegetarian (u), căsătorit(z, x).
Să se scrie propoziţiile:
a.Toţi oamenii, cu excepţia măcelarilor, îi iubesc pe vegetarieni.
b. Unele femei vegetariene sunt căsătorite cu măcelari.

27. Se dau predicatele: iubeşte(x, z); om(x); măcelar(z), femeie (w), vegetarian (u), căsătorit(z, x).
Să se scrie propoziţiile:
a. Niciun vegetarian nu-i iubeşte pe vegetarieni.
b. Unii bărbaţi necăsătoriţi sunt măcelari.

28. Se dau predicatele: iubeşte(x, z); om(x); măcelar(z), femeie (w), vegetarian (u), căsătorit(z, x).
Să se scrie propoziţiile:
a. Soţiile vegetarienilor îi iubesc pe vegetarieni.
b. Nevegetarienii îi iubesc pe măcelari.

29. Se dau predicatele: iubeşte(x, z); om(x); măcelar(z), femeie (w), vegetarian (u), căsătorit(z, x).
Să se scrie propoziţiile:
a. Toate femeile necăsătorite iubesc măcelarii.
b. Toţi măcelarii vegetarieni sunt femei.
Se dau predicatele: iubeşte(x, z); om(x); măcelar(z), femeie (w), vegetarian (u), căsătorit(z, x).

30. Să se traducă în limba română formulele:
a. $\forall w((\text{femeie}(w) \wedge \neg\exists x \text{ căsătorit}(x, w)) \supset (\forall z (\text{măcelar}(z) \supset \text{iubeşte}(w, z)))$
b. $\forall z((\text{măcelar}(z) \wedge \text{vegetarian}(z)) \supset \text{femeie}(z))$

31. Se dau predicatele: ciupercă(x); roşu (u); otrăvitor(y), angajează (x, y) , mănâncă(x, y) .
Să se traducă în limba română formulele:
a. $\forall x((\text{ciupercă}(x) \wedge \text{roşu}(x)) \supset \text{otrăvitor}(u))$
b. $\forall x (\exists y(\text{angajează}(x, y)) \supset \forall z(\text{ciupercă}(z) \supset \neg\text{mănâncă}(x, z)) \wedge \exists u \exists z(\text{angajează}(u, t) \wedge \text{mănâncă}(t, z) \wedge \text{ciupercă}(z))$

32. Se dă predicatul S(x, y, z) = „x spune lui y, z"
Definiţi predicatele:
a. duhovnic(x) (i.e. persoana căruia îi spun toţi toate);
b. gură spartă (x) (i.e. persoana care spune tuturor toate).

33. Se dă predicatul S(x, y, z) = „x spune lui y, z"
Să se defmească:
a. predicatul binar S1(x, y) = „x spune totul lui y";
b. predicatul unar S2(x) = „x spune cuiva orice"

34. Se dau predicatele: D(x) = x este Dumnezeu; B(x, y) = „x bate pe y"; B1(y) = „y este o bâtă".
Să se formalizeze propoziţia: „Dumnezeu nu bate cu bâta!"

35. Se dau predicatele C(x) = „X este un cui"; S(x, y) = „x scoate afară pe y"
Să se formalizeze în limbajul logicii predicatelor proverbul: „Cui pe cui se scoate afară"

D. *EXERCIȚII DE DECODIFICARE*

1. Se dă formula bine formată $\forall x \exists y M(y, x)$ în limbajul logicii predicatelor și predicatul M(y, x) = „y este mama lui x"
Să se redea în limba română semnificația formulei.

2. Se dă formula bine formată $\forall z \forall v(M(z, v) \supset I(z, v))$ în limbajul logicii predicatelor și predicatele M(z, v) = „z este mama lui v" și I(z, v) = „z iubește pe v"
Să se exprime în limba română înțelesul formulei.

3. Se dă formula bine formată în limbajul logicii predicatelor: $\forall t(\exists u I(u, t) \supset V(t))$ și predicatele: I(u, t) = „u iubește pe t" și V(t) = „t este vesel"
Să se redea în limba română semnificația formulei.

4. Se dă formula bine formată în limbajul logicii predicatelor:
$\forall x\{ G(x) \supset \exists y[S1(x) \wedge (F(x, y) \wedge D(y)]\}$
și predicatele: G(x) = x este generos; S1(z) = z este sensibil; S(v) = v este stimat; F(s, t) = s face t; C(z) = z este competent; D(t) = t este un dar.
Să se redea în limba română semnificația formulei.

5. Se dă formula bine formată în limbajul logicii predicatelor: $\exists z($ C(z) $\wedge$G(z))
și predicatele: C(z) = z este competent G(x) = x este generos.
Să se redea în limba română semnificația formulei.

6. Se dă formula bine formată în limbajul logicii predicatelor: $\forall s[\exists t(F(s, t) \wedge D(t)) \supset S(s)]$
și predicatele: F(s, t) = s face t; C(z) = z este competent; D(t) = t este un dar.
Să se redea în limba română semnificația formulei.

7. Se dă formula bine formată în limbajul logicii predicatelor: $\forall x [V(x) \supset (\forall y(A(x, y) \wedge (O(y, x) \supset B(y))$ şi predicatele: V(x) = „x este vanitos"; B(z) = „z este bun"; A(x, y) = „x are y"
O (y, x) = „x este o opinie despre y".
Să se redea în limba română semnificaţia formulei.

8. Se dă formula bine formată în limbajul logicii predicatelor:
$\forall z ((P(z) \supset V(z)) \wedge \forall z(P(z) \supset \forall w(E(w, z) \supset I(z, w))$
şi predicatele: P(u) = „u este politician"; V(x) = „x este vanitos"; E(t, u) = „t este un elogiu despre u" I(x, y) = „x iubeşte pe y".
Să se redea în limba română semnificaţia formulei.

9. Se dă formula bine formată în limbajul logicii predicatelor:
$\forall s[M(s) \supset \neg\exists t(A(s, t) \wedge O(t, s) \wedge B(t)] \wedge \forall s (M(s) \supset \forall u(E(u, s) \supset \neg I(s,u))]$ şi predicatele: V(x) = „x este vanitos"; B(z) = „z este bun"; M(s) = „s este modest"; P(u) = „u este politician"; E(t, u) = „t este un elogiu despre u"; I(x, y) = „x iubeşte pe y"; A(x. y) = „x are y"; O (y, x) = „x este o opinie despre y"
Să se redea în limba română semnificaţia formulei.
10. Se dau formulele bine formate în limbajul logicii predicatelor:
1. $\forall x (F(x) \supset \neg A(x))$, 2. $\forall y (P(y) \supset F(y))$; 3. $\exists z (R(z) \wedge A(z))$
şi predicatele: F(x) = „x este pom fructifer"; A(x) = „x este arbust alpin"; P(y) = „y este un piersic";
R(z) = „z este un răşinos".
Să se redea în limba română semnificaţia formulelor.

11. Se dau formulele bine formate în limbajul logicii predicatelor:
1. $\forall x (C(x) \supset \neg I(x))$; 2. $\forall y(H(y) \supset I(y))$; 3. $\exists z(H(z) \wedge S(z))$
şi predicatele: C(x) = „x este om cinstit"; I(x) = „x este un infractor"; H(y) = „y este un hoţ";
S(z) = „z este un sentimental"
Să se redea semnificaţia lor în limba română.

12. Se dă formula bine formată în limbajul logicii predicatelor:
$\forall x \forall y((H(x) \wedge C(y)) \supset R(x, y))$
şi predicatele: H(x) = „x este un cal"; C(x) = „x este un câine"; R(x, y) = „x este mai rapid decât y"
Să se redea în limba română semnificaţia formulei.

13. Se dă formula bine formată în limbajul logicii predicatelor:
$\exists u\ (O(u) \wedge \forall t(I(t) \supset R(u, t))$
şi predicatele: O(y) = „y este un ogar"; I(y) = „y este un iepure"; R(x, y) = „x este mai rapid decât y"
Să se redea în limba română semnificaţia formulei.

14. Se dă formula bine formată în limbajul logicii predicatelor:
$\forall s \forall v \forall w\ ((R(s, v) \wedge R(v, w)) \supset R(s, w))$
şi predicatul: R(x, y) = „x este mai rapid decât y".
Să se redea în limba română semnificaţia formulei.

15. Se dă formula bine formată în limbajul logicii predicatelor:
$\forall x[(S(x) \vee D(x)) \supset L(x)]$
şi predicatele: S(x) = „x este senator"; D(x) = „x este deputat"; L(x) = „x este legiuitor.
Să se redea în limba română semnificaţia formulei.

16. Se dă formula bine formată în limbajul logicii predicatelor:
$\forall y \forall z[(Z(y) \wedge P(y, z) \supset V(z)) \supset \exists t \exists u\ (S(u) \wedge F(u, t) \wedge I(t))]$
şi predicatele: S(x) = „x este senator"; Z(y) = „y este ziarist"; P(y, z) = „y publică z"; V(z) = „z este adevărat"; F(s, t) = „u a săvârşit t"; I(t) = „t este o infracţiune".
Să se redea în limba română semnificaţia formulei.

17. Se dă formula bine formată în limbajul logicii predicatelor:
$\forall s[\exists v(F(s, v) \wedge I(v)) \supset C(s)]$
şi predicatele: F(s, t) = „u a săvârşit t"; I(t) = „t este o infracţiune";
C (t) = „t este compromis
Să se redea în limba română semnificaţia formulei.

18. Se dau formulele bine formate în limbajul logicii predicatelor:
$\forall x(D(x) \supset L(x))$
$\exists y \exists z(D(y) \wedge S(y, z) \wedge I(z))$
şi predicatele: D(x) = „x este un deputat"; L(x) = „x este un legiuitor"; S(y, z) = „y săvârşeşte z"; I(z) = „z este o infracţiune"
Să se redea în limba română semnificaţia formulelor.

19. Se dă formula bine formată în limbajul logicii predicatelor:
$\forall u[\exists t(S(u, t) \wedge I(t)) \supset \exists s(\neg R(u, s) \wedge L1(s))]$

şi predicatele: S(y, z) = „y săvârşeşte z"; I(z) = „z este o infracţiune"; R(u, t) = „u respectă t"; Ll(t) = „t este o lege"
Să se redea în limba română semnificaţia formulei.

20. Se dă formula bine formată în limbajul logicii predicatelor:
$\forall x(P(x) \supset N(x))$, $\exists y(N(y) \wedge I(y))$
şi predicatele: P(x) = „x este un politician";
N(x) = „x este nestatornic"; I(x) = „x este ipocrit".
Să se redea în limba română semnificaţia formulei.

21. Se dă formula bine formată în limbajul logicii predicatelor:
$\forall x(F(x) \supset -U(x))$; $\exists y(U(y) \wedge GA(y))$
şi predicatele: F(x) = x este folositor; GA (x) = „x este general accesibil"
Să se redea semnificaţia formulei în limba română.

22. Se dă formula bine formată în limbajul logicii predicatelor: $\exists xS(x, b) \supset S(a, b)$ şi predicatul: S(x, y) = „x sare y" şi constantele individuale: a = Ştefan, b = acest obstacol.
Să se redea în limba română semnificaţia formulei.

23. Se dă formula bine formată în limbajul logicii predicatelor:
$L(a) \supset (\exists y(C(a, y) \supset N(a, y))$ şi predicatele: L(x) = „x învaţă";
C((x, y) = „x este căsătorit cu y"; N(x, y) = „x neglijează pe y"
Să se redea în limba română semnificaţia formulei.

24. Se dă formula bine formată în limbajul logicii predicatelor:
$\forall z\ [\exists u(C(u, z) \wedge N(u, z)) \supset (-F(z) \vee I(z))]$
şi predicatele: C(x, y) = „x este căsătorit cu y";
N(x, y) = „x neglijează pe y";
I(x) = „x este infidel"; F(x) = „x este fericit"
Să se redea în limba română semnificaţia formulei.

25. Se dă formula bine formată în limbajul logicii predicatelor:
$\exists y(C(a, y) \wedge F(y))$
şi predicatele: C((x, y) = „x este căsătorit cu y"; F(x) = „x este fericit";
a = Ştefan.
Să se redea în limba română semnificaţia formulei.

26. Se dă formula bine formată în limbajul logicii predicatelor: $\forall v[\exists t(C(v, t) \wedge I(t)) \supset C1(v)]$
şi predicatele: C(x, y) = „x este căsătorit cu y"; I(x) = „x este infidel"; C1(x) = „x este încornorat".
Să se redea semnificaţia formulei în limba română.

27. Se dau formulele bine formate în limbajul logicii predicatelor: $\exists xT(x) \wedge \exists yM(y)$, $\forall z[(T(z) \vee M(z)) \supset R(z)]$
şi predicatele: T(x) = „x este un trandafir"; M(y) = „este un măceş"; R(v) = „v este o rozacee"
Să se redea în limba română semnificaţia formulei.

28. Se dă formula bine formată în limbajul logicii predicatelor: .$\exists xT(x) \wedge \exists yM(y)$ şi predicatele: T(x) = „x este un trandafir"; M(y) = „este un măceş".
Să se redea în limba română semnificaţia formulelor.

29. Se dă formula bine formată în limbajul logicii predicatelor: $\forall z[(T(z) \vee M(z)) \supset R(z)]$
şi predicatele: T(x) = „x este un trandafir"; M(y) = „este un măceş"; R(v) = „v este o rozacee".
Să se redea în limba română semnificaţia formulelor.
30. Se dau formulele bine formate în limbajul logicii predicatelor: $\forall u[M(u) \supset (S(u) \wedge C(u))]$, $\forall t(T(t) \supset C1(t))$
şi predicatele:; M(y) = „este un măceş"; S(v) = „v este sălbatec"; C(z) = „z este comestibil"; T(x) = „x este un trandafir"; C1(w) = „w este un arbust cultivabil".
Să se redea în limba română semnificaţia formulei.

31. Se dă formula bine formată în limbajul logicii predicatelor: $\forall x(E(x) \supset \exists y(A(y) \wedge P(x, y))$
şi predicatele: E(x) = „x este excroc"; A(y) = „y este ageamiu"; P(x, y) = „x păcăleşte pe y".
Să se redea în limba română semnificaţia formulei.

32. Se dau formulele bine formate în limbajul logicii predicatelor: $\forall z(S(z) \supset E(z))$, $\exists u(S(u) \wedge B(u))$
şi predicatle: E(x) = „x este excroc"; A(y) = „y este ageamiu";

I(u) = „u este înalt";
B(x) = „x este bucureştean; S(z) = „z este samsar";
P(x, y) = „x păcăleşte pe y"
Să se redea în limba română semnificaţia formulelor.

33. Se dă formula bine formată în limbajul logicii predicatelor:
$\forall v\,(A(v) \wedge \exists t(S(t) \wedge P(t, v)) \supset I(v))$
şi predicatele: E(x) = „x este excroc"; A(y) = „y este ageamiu";
I(u) = „u este înalt"; S(z) = „z este samsar";
P(x, y) = „x păcăleşte pe y"
Să se redea în limba română semnificaţia formulei.

34. Se dă formula bine formată în limbajul logicii predicatelor:
$\forall x[\exists y(C(x, y) \wedge P(y) \wedge \neg S(x, y)) \supset G(x)]$
şi predicatele: G(x) = „x este un geniu"; C(x, y) = „x cunoaşte pe y";
P(u) = „u este un lucru profund"; S(x, y) = „x studiază temeinic y"
Să se redea în limba română semnificaţia formulei.

35. Se dau formulele bine formate în limbajul logicii predicatelor:
$\forall z\forall t(S(z, t) \supset C(z, t))$, $\forall s\forall r[(S1(s, r) \wedge \neg C(s, r)) \supset (D(s) \vee IM(s))]$
şi predicatele: D(x) = „x este un diletant";IM(x) = „x este un impostor";Z(x) = „x este un ziarist"; C(x, y) = „x cunoaşte pe y";
S(x, y) = „x studiază temeinic y"; S1(x, y) = „x scrie despre y"
Să se redea în limba română semnificaţia formulelor.
36. Se dă formula bine formată în limbajul logicii predicatelor:
$\exists u\exists v(Z(u) \wedge S1(u, v) \wedge \neg C(u, v))$
şi predicatele: Z(x) = „x este un ziarist"; C(x, y) = „x cunoaşte pe y";
S(x, y) = „x studiază temeinic y"; S1(x, y) = „x scrie despre y"
Să se redea în limba română semnificaţia formulei.

37. Se dă formula bine formată în limbajul logicii predicatelor:
$\exists m\exists n(Z(m) \wedge C(m, n) \wedge P(n) \wedge \neg S(m, n))$
şi predicatele: Z(x) = „x este un ziarist"; C(x, y) = „x cunoaşte pe y";
S(x, y) = „x studiază temeinic y"; S1(x, y) = „x scrie despre y"
Să se redea în limba română semnificaţia formulei.

38. Se dă formula bine formată în limbajul logicii predicatelor: $\forall x\forall y$
$[(M(x) \wedge E(x, y)) \supset D(x, y)]$

şi predicatele: M(x) = „x este un medic competent"; E(x, y) = „x examinează atent pe y"; D(x, y) = „x pune lui y un diagnostic corect". Să se redea în limba română semnificaţia formulei.

39. Se dă formula bine formată în limbajul logicii predicatelor: $\forall z \forall u\ (M(z) \wedge -E(z, u)) \supset (O(z) \vee F(z))$
şi predicatele: : M(x) = „x este un medic competent";
E(x, y) = „x examinează atent pe y"; O(x) = „x este obosit";
F(x) = „x este lipsit de etică profesională".
Să se redea în limba română semnificaţia formulei.

40. Se dă formula bine formată în limbajul logicii predicatelor: $\forall t(T(t, b) \supset M(t))$
şi predicatele: M(x) = „x este un medic competent"; T(x, y) = „x şi-a luat doctoratul cu y"
Să se redea în limba română semnificaţia formulei.

41. Se dă formula bine formată în limbajul logicii predicatelor: $\forall x[(T(x) \wedge F(x)) \supset P(x)]$
şi predicatele: T(x) = „x este tânăr"; F(x) = „x este frumos";
P(x) = „x este plăcut".
Să se redea în limba română semnificaţia formulei.

42. Se dau formulele bine formate în limbajul logicii predicatelor: $\forall y(P(y) \supset I(y))$, $\forall z(I(z) \supset C(z))$
şi predicatele: P(x) = „x este plăcut"; I(x) = „x este invidiat";
C(x) = „x este criticat"
Să se redea în limba română semnificaţia formulelor.

43. Se dă formula bine formată în limbajul logicii predicatelor: $\forall u[(L(u) \vee P1(u)) \supset C(u)]$
şi predicatele: C(x) = „x este criticat";
L(x) = „x este luat în consideraţie";
P1(x) = „x este performant".
Să se redea în limba română semnificaţia formulei.

44. Se dă formula bine formată în limbajul logicii predicatelor: $(-T(a) \vee -F(a)) \wedge -L(a) \wedge -P1(a)$
şi predicatele: T(x) = „x este tânăr"; F(x) = „x este frumos";
L(x) = „x este luat în consideraţie"; P1(x) = „x este performant"
Să se redea în limba română semnificaţia formulei.

45. Se dă formula bine formată în limbajul logicii predicatelor: $\forall x \forall y$ ($(C(x) \wedge S(y)) \supset I(x, y)$)
şi predicatele: (C(x) = „x este catolic";S(y) = „x este supus";
I(x, y) = „x iubeşte pe y".
Să se redea în limba română semnificaţia formulei.

46. Se dă formula bine formată în limbajul logicii predicatelor: $\forall z \forall u$ ($(C(z) \wedge A(u)) \supset -I(z, u)$
şi predicatele: (C(x) = „x este catolic"; A(x) = „x este ateu";
I(x, y) = „x iubeşte pe y".
Să se redea în limba română semnificaţia formulei..

47. Se dă formula bine formată în limbajul logicii predicatelor:
$\forall v (S(v) \supset -A(v))$
şi predicatele: S(y) = „x este supus"; A(x) = „x este ateu"
Să se redea în limba română semnificaţia formulei.

48. Se dau formulele bine formate în limbajul logicii predicatelor:
$\forall x(G(x) \supset P(x))$, $\forall y(M(y) \supset -G(y))$
şi predicatele: G(x) = „x este generos"· P(x) = „x este puternic";
M(x) = „x este meschin".
Să se redea în limba română semnificaţia formulelor.

49. Se dă formula bine formată în limbajul logicii predicatelor:
$\forall x \forall y(-S(x, y) \supset -C(x, y))$
şi predicatele: C(x, y) = „x cunoaşte y"; S(x, y) = „x a studiat y"
Să se redea în limba română semnificaţia formulei.

50. Se dă formula bine formată în limbajul logicii predicatelor:
$\forall z \forall u(Z(z) \supset [S1(z, u) \wedge (E(u) \vee D(u) \vee P(u) \vee A(u) \vee K(u))]$
şi predicatele: S1(z, u) = „z scrie despre u"; E(t) = „t este un subiect de economie";
D(t) = „t este un subiect de drept"; A(t) = „este un subiect de artă";
K(t) = „t este un subiect de ştiinţă";
Să se redea în limba română semnificaţia formulei.

51. Se dă formula bine formată în limbajul logicii predicatelor:
$\forall t \forall v \{[Z(t) \wedge (E(v) \vee D(v) \vee P(v) \vee A(v) \vee K(v))] \supset -S(t, v)\}$
şi predicatele: Z(v) = „v este ziarist"; E(t) = „t este un subiect de economie"; D(t) = „t este un subiect de drept"; P(t) = „t este un subiect

de politică"; C(x, y) = „x cunoaşte y";K(t) = „t este un subiect de ştiinţă"; S(x, y) = „x a studiat y".
Să se redea în limba română semnificaţia formulei.

52. Se dă formula bine formată în limbajul logicii predicatelor: $\forall w(Z(w) \supset \exists s(S1(w, s) \wedge \neg C(w, s)))$
şi predicatele: Z(v) = „v este ziarist"; S1(z, u) = „z scrie despre u"; C(x, y) = „x cunoaşte y".
Să se redea în limba română semnificaţia formulei.

53. Se dă formula bine formată în limbajul logicii predicatelor:
$(B(a) \& D(a)) \supset \forall x(F(x) \supset I(x, a))$
şi predicatele: B(x) = „x este frumos"; D(x) = „x dansează";
F(x) = „x este o fată"; I(x, y) = „x se îndrăgosteşte (iubeşte) pe y"
Să se redea în limba română semnificaţia formulei.

54. Se dă formula bine formată în limbajul logicii predicatelor: $\forall y \forall z \neg[B(a, y, z) \wedge P(y) \wedge B1(z)]$
şi predicatele D(x) = x este Dumnezeu; B(x, y) = „x bate pe y" B1(y) = „y este o bâtă"
Să se exprime semnificaţia formulei în limba română.
Construiţi un exerciţiu propriu similar !.

E. *SUBIECTE DE SILOGISTICĂ REZOLVATE PRIN LOGICA PREDICATELOR*

1. Se dau silogismele: Dacă MaP şi SoM, atunci SiP şi Dacă PaM şi SeM, atunci SeP.
Să se decidă asupra validităţii lor prin:
a) diagrame Euler;
b) formalizare în logica predicatelor şi aplicarea metodei arborilor de decizie.

2. Se dau silogismele: Dacă MaP şi SaM, atunci SiP şi Dacă MeP şi SiM, atunci SoP.
Să se decidă asupra validităţii lor prin:
a) diagrame Venn;
b) formalizare în logica predicatelor şi aplicarea metodei rezoluţiei.

3. Se dau silogismele: Dacă MiP şi MaS, atunci SiP şi Dacă PeM şi SaM, atunci SeP.
Să se decidă asupra validităţii lor prin:
a) diagrame Venn şi metoda Davis-Putnam;
b) formalizare în logica predicatelor şi aplicarea metodei arborilor de decizie.

4. Se dau silogismele: Dacă PaM şi MiS, atunci SiP şi Dacă MaP şi Mi S, atunci SiP.
Să se decidă asupra validităţii lor prin:
a) diagrame Venn;
b) formalizare în logica predicatelor şi aplicarea metodei rezoluţiei.

5.Se dau silogismele: Dacă PaM şi MiS, atunci SiP şi Dacă MeP şi SiM, atunci SoP.
Să se decidă asupra validităţii lor prin:
a) diagrame Euler;
b) formalizare în logica predicatelor şi aplicarea metodei tablourilor semantice.

6. Se dau silogismele: Dacă MaP şi MiS, atunci SiP şi Dacă MeP şi SiM, atunci SoP.
Să se decidă asupra validităţii lor prin:
a) diagrame Venn;
b) formalizare în logica predicatelor şi aplicarea metodei arborilor de decizie.

7. Se dau silogismele: Dacă PeM şi SoM, atunci SoP şi Dacă MaP şi SiM, atunci SiP
Să se decidă asupra validităţii lor prin:
a) diagrame Euler;
b) formalizare în logica predicatelor şi aplicarea metodei rezoluţiei.

8. Se dau silogismele: Dacă PeM şi MiS, atunci SiP şi Dacă PaM şi SiP, atunci SiP
Să se decidă asupra validităţii lor prin:
a) diagrame Euler;
b) formalizare în logica predicatelor şi aplicarea metodei Davis-Putnam.

9. Se dau silogismele: Dacă PaM şi MoS, atunci SoP şi Dacă MaP şi SiM, atunci SiP
Să se decidă asupra validităţii lor prin:
a) diagrame Karnaugh;
b) formalizare în logica predicatelor şi aplicarea metodei tablourilor semantice.

10. Se dau silogismele: Dacă PeM şi MiS, atunci SiP şi Dacă MeP şi SiM, atunci SoP.
Să se decidă asupra validităţii lor prin:
a) diagrame Venn;
b) formalizare în logica predicatelor şi aplicarea metodei rezoluţiei semantice .

11. Se dau silogismele:Dacă PaM şi SiM, atunci SiP şi Dacă MeP şi SoM, atunci SoP
Să se decidă asupra validităţii lor prin:
a) arbori de decizie;
b) formalizare în logica predicatelor şi aplicarea metodei Davis-Putnam.

12. Se dau silogismele: PeM şi SaM, atunci SeP şi Dacă MaP şi SiM, atunci SiP. Să se decidă asupra validităţii lor prin
a) metoda rezoluţiei strategia priorităţii clauzei unitare;
b) formalizare în logica predicatelor şi aplicarea metodei arborilor de decizie.

13. Se dau silogismele: Dacă PeM şi SiM, atunci SoP şi Dacă PaM şi MiS, atunci SiP.
Să se decidă asupra validităţii lor prin:
a) diagrame Karnaugh;
b) formalizare în logica predicatelor şi aplicarea metodei arborilor de decizie.

14. Se dau silogismele: Dacă PiM şi SaM, atunci SiP şi Dacă PeM şi SaM,atunci SiP
Să se decidă asupra validităţii lor:
a) diagrame Euler;
b) formalizare în logica predicatelor şi aplicarea metodei rezoluţiei.

15. Se dau silogismele: PeM şi Sa M, atuci SiP şi Dacă PeM şi SaM, atunci SiP.
Să se decidă asupra validităţii lor prin:
a) diagrame Venn;
b) formalizare în logica predicatelor şi aplicarea metodei arborilor de decizie.

16. Se dau silogismele: MaP şi MiS, atunci SiP şi Dacă PeM şi SaM, atunci PeM.
Să se decidă asupra validităţii lor prin:
a) diagrame Euler;
b) formalizare în logica predicatelor şi aplicarea metodei rezoluţiei semantice.

17. Se dau silogismele: Dacă PiS şi SaM, atunci SiP şi Dacă MeP şi SaM, atunci SeP.
Să se decidă asupra validităţii lor prin:
a) diagrame Karnaugh;
b) formalizare în logica predicatelor şi aplicarea metodei rezoluţiei semantice.

18 Se dau silogismele: Dacă PaM şi SiM, atunci SiP şi Dacă PeM şi SiM, atunci SiP.
Să se decidă asupra validităţii lor:
a) diagrame Venn;
b) formalizare în logica predicatelor şi aplicarea metodei arborilor de decizie.

19. Se dau silogismele: Dacă PeM şi SaM, atunci SeP şi Dacă MaP şi MaS, atunci SiP.
Să se decidă asupra validităţii lor prin:
a) diagrame Venn;
b) formalizare în logica predicatelor şi aplicarea metodei arborilor de decizie.

20 Se dau silogismele: Dacă PeM şi SaM, atunci SiP şi Dacă MeP şi SiP, atunci SoP
Să se decidă asupra validităţii lor prin:
a) diagrame Euler;
b) formalizare în logica predicatelor şi aplicarea metodei arborilor de decizie.

21. Se dau silogismele: Dacă PaM şi MiS, atunci SiP şi Dacă MeP şi MiS, atunci SiP.
Să se decidă asupra validităţii lor prin:
a) diagrame Euler;
b) formalizare în logica predicatelor şi aplicarea metodei rezoluţiei.

22. Se dau silogismele: Dacă PaM şi MiS, atunci SiP şi Dacă PeM şi SaM, atunci SiP.
Să se decidă asupra validităţii lor prin:
a) diagrame Euler;
b) formalizare în logica predicatelor şi aplicarea metodei rezoluţiei strategia de input.

23. Se dau silogismele: Dacă MeP şi Si M, atunci SiP şi Dacă PeM şi SiM, atunci SoM.
Să se decidă asupra validităţii lor prin:
a) diagramelor Venn;
b) formalizare în logica predicatelor şi aplicarea metodei tablourilor semantice.

24. Se dau silogismele: Dacă PeM şi SaM, atunci SiP şi Dacă MaP şi MiS, atunci SiP
Să se decidă asupra validităţii lor prin:
a) diagrame Euler;
b) formalizare în logica predicatelor şi aplicarea metodei arborilor de decizie.

25. Se dau silogismele: Dacă MeP şi SaM, atnci SeP şi dacă PeM şi MiS, atunci SiP.
Să se decidă asupra validităţii lor prin:
a) diagrame Karnaugh;
b) formalizare în logica predicatelor şi aplicarea metodei rezoluţiei semantice.

26. Se dau silogismele: Dacă PeM şi SaM, atunci SeP şi Dacă PaM şi SiM, atunci SiP.
Să se decidă asupra validităţii lor:
a) diagrame Venn;
b) formalizare în logica predicatelor şi aplicarea metodei Davis-Putnam.

27. Se dau silogismele: Dacă PeM şi MaS, atunci SeP şi Dacă MeP şi SaM, atunci SoP.
Să se decidă asupra validităţii lor prin:
a) diagrame Euler;
b) formalizare în logica predicatelor şi aplicarea metodei arborilor de decizie

28 Se dau silogismele· PaM şi SiM, atunci SiP şi Dacă PeM şi SaM, atunci SiP
Să se decidă asupra validităţii lor prin:
a) diagrame Venn;
b) formalizare în logica predicatelor şi aplicarea metodei rezoluţiei.

29. Se dau silogismele: Dacă PaM şi MiS, atunci SiP şi Dacă PeM şi SaM, atunci SiP
Să se decidă asupra validităţii lor:
a) diagrame Euler;
b) formalizare în logica predicatelor şi aplicarea metodei rezoluţiei strategia de input.

30. Se dau silogismele: Dacă PeM şi MaS, atunci SeP şi Dacă PeM şi SiP, atunci SoP.
Să se decidă asupra validităţii lor prin:
a) diagrame Euler;
b) formalizare în logica predicatelor şi aplicarea metodei rezoluţiei semantice.

31. Se dau silogismele: Dacă MaP şi SaM, atunci SiP şi Dacă PeM şi SiM, atunci Sip
Să se decidă asupra validităţii lor prin:
a) diagrame Karnaugh;
b) formalizare în logica predicatelor şi aplicarea metodei respingerii rezolutive.

32. Se dau silogismele: Dacă PeM şi SaM, atunci SiP şi dacă PeM şi MiS, atunci SiP.
Să se decidă asupra validităţii lor prin:
a) metoda diagramelor Euler;
b) formalizare în logica predicatelor şi aplicarea metodei.

33. Se dau silogismele: Dacă PaM şi SiM, atunci SiP şi Dacă PeM şi SiM , atunci SoP.
Să se decidă asupra validităţii lor prin:
a) diagrame Venn;
b) formalizare în logica predicatelor şi aplicarea metodei arborilor de decizie.

34. Se dau silogismele: Dacă PeM şi MiS, atunci SoP şi dacă MeP şi SaM, atunci SoP.
Să se decidă asupra validităţii lor prin:
a)diagrame Karnaugh;
b) formalizare în logica predicatelor şi aplicarea metodei respingerii rezolutive.

35. Se dau silogismele: Dacă OeM şi SiM, atunci SiP şi Dacă MaP şi SiP, atunci SiP.
Să se decidă asupra validităţii lor prin:
a) diagrame Euler;
b) formalizare în logica predicatelor şi aplicarea metodei rezoluţiei liniare.

36. Se dau silogismele: Dacă PeM şi MaS, atunci SeP şi Dacă PeM şi SiM, atunci SoP.
Să se decidă asupra validităţii lor prin:
a) diagrame Venn;
b) formalizare în logica predicatelor şi aplicarea metodei respingerii rezolutive.

37. Se dau silogismele: Dacă PaM şi SiM, atunci SiP şi Dacă PeM şi SaM , atunci SiP.
Să se decidă asupra validităţii lor:
a) diagrame Euler;
b) formalizare în logica predicatelor şi aplicarea metodei rezoluţiei semantice.

38. Se dau silogismele: Dacă MaP şi SaM, atunci SaP şi Dacă PaM şi SaM, atunci SiP.
Să se decidă asupra validităţii lor prin:
a) diagrame Euler;
b) formalizare în logica predicatelor şi aplicarea metodei respingerii rezolutive.

39. Se dau silogismele: Dacă PaM şi MaS, atunci SiP şi Dacă PeM şi M iS, atunci SiP.
Să se decidă asupra validităţii lor:
a) diagrame Venn;
b) formalizare în logica predicatelor şi aplicarea metodei respingerii rezolutive.

40. Se dau silogismele: Dacă MeP şi SaM, atunci SoP şi Dacă MeP şi SaM , atunci SiP.
Să se decidă asupra validităţii lor prin:
a) diagrame Euler;
b) formalizare în logica predicatelor şi aplicarea metodei rezoluţiei liniare.

41. Se dau silogismele: Dacă PaM şi SiM, atunci SiP şi Dacă PeM şi MaS, atunci SiP.
Să se decidă asupra validităţii lor prin:
a) diagrame Euler;
b) formalizare în logica predicatelor şi aplicarea metodei respingerii rezolutive.

42. Se dau silogismele: Dacă PeM şi SaM, atnci SeP şi dacă PaM şi SiM, atunci SiP.
Să se decidă asupra validităţii lor prin:
a) diagrame Venn;
b) formalizare în logica predicatelor şi aplicarea metodei rezoluţiei.

43. Se dau silogismele: Dacă PeM şi SaM, atunci SeP şi Dacă PaM şi SaM, atunci SeP.
Să se decidă asupra validităţii lor prin:
a) diagrame Venn;
b) formalizare în logica predicatelor şi aplicarea metodei arborilor de decizie.

44. Se dau silogismele: Dacă PeM şi MiS, atunci SeP şi Dacă MaP şi SaM, atunci SaM.
Să se decidă asupra validităţii lor prin:
a) diagrame Euler;
b) formalizare în logica predicatelor şi aplicarea metodei tablourilor semantice.

45. Se dau silogismele: Dacă PeM şi SaM, atunci SiP şi Dacă MaP şi MiS, atunci SiP.
Să se decidă asupra validităţii lor prin:
a) diagrame Karnaugh;
b) formalizare în logica predicatelor şi aplicarea metodei Davis-Putnam.

46. Se dau silogismele:Dacă MaP şi SiM, atunci SiP şi Dacă PeM şi MaS, atunci SiP.
Să se decidă asupra validităţii lor prin:
a) diagrame Venn;
b) formalizare în logica predicatelor şi aplicarea metodei

47. Se dau silogismele: Dacă MeP şi SiM, atunci siP şi Dacă PaM şi SiM, atunci SoP.
Să se decidă asupra validităţii lor prin:
a) diagrame Euler;
b) formalizare în logica predicatelor şi aplicarea metodei rezoluţiei strategia rezoluţiei liniare.

48. Se dau silogismele: Dacă MeP şi MaS, atunci SiP şi dacă PaM şi MiS, atunci SiP.
Să se decidă asupra validităţii lor prin:
a) diagrame Karnaugh;
b) formalizare în logica predicatelor şi aplicarea metodei respingerii rezolutive.

49. Se dau silogismele: Dacă PaM şi SaM, atunci SiP şi dacă PaM şi MiS, atunci SiP.
Să se decidă asupra validităţii lor prin:
a) diagrame Venn;
b) formalizare în logica predicatelor şi aplicarea metodei arborilor de decizie.

50 . Se dau silogismele: Dacă PeM şi SaM, atunci SiP şi dacă PaM şi MiS, atunci SiP.
Să se decidă asupra validităţii lor prin:
a) diagrame Euler;
b) formalizare în logica predicatelor şi aplicarea metodei Davis-Putnam.

CENTRALĂ UNIV

LUCRĂRI APĂRUTE
ÎN EDITURA FUNDAȚIEI *ROMÂNIA DE MÂINE*

Ion Tudosescu
LUCIAN BLAGA. CONCEPȚIA ONTOLOGICĂ

Ioan N. Roşca
FILOSOFIA MODERNĂ

C Rădulescu - Motru
ETNICUL ROMÂNESC

Alexandru Surdu
CONTRIBUȚII ROMÂNEŞTI ÎN DOMENIUL LOGICII
ÎN SECOLUL XX

Ioan C Ivanciu
FILOSOFIA ISTORIEI

Acsinte Dobre
INTRODUCERE ÎN EPISTEMOLOGIE

Ion Tudosescu
IDENTITATEA AXIOLOGICĂ A ROMÂNILOR

Şerban C. Andronescu
ANALOGII ESTETICE. TEORIA SPAȚIULUI - TIMP ÎN ARTE
ŞI LITERATURĂ

Georgeta Marghescu
INTRODUCERE ÎN ANTROPOLOGIA CULTURALĂ

ISBN 973—582—255— 5

Preț 95 000 Lei

EDITURA FUNDAȚIEI *ROMÂNIA DE MÂINE*